2016

# A Hybrid Approach to Aerial Video Image Registration

Karol T. Salva
*Wright State University*

# A HYBRID APPROACH TO AERIAL VIDEO IMAGE REGISTRATION

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

by

KAROL T. SALVA
B.S. C.S.E., Ohio State University, 2009

2016
Wright State University

Wright State University
GRADUATE SCHOOL

January 23, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY KAROL T. SALVA ENTITLED A HYBRID APPROACH TO AERIAL VIDEO IMAGE REGISTRATION BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

Arthur A. Goshtasby, Ph.D.
Thesis Director

Mateen M. Rizki, Ph.D.
Chair, Department of Computer
Science and Engineering

Committee on
Final Examination

Arthur A. Goshtasby, Ph.D.

Thomas Wischgoll, Ph.D.

Juan R. Vasquez, Ph.D.

Robert E.W. Fyffe, Ph.D.
Vice President for Research and
Dean of the Graduate School

# ABSTRACT

SALVA, KAROL T. M.S., Department of Computer Science and Engineering, Wright State University, 2016. *A HYBRID APPROACH TO AERIAL VIDEO IMAGE REGISTRATION.*

Many video processing applications, such as motion detection and tracking, rely on accurate and robust alignment between consecutive video frames. Traditional approaches to video image registration, such as pyramidal Kanade-Lucas-Tomasi (KLT) feature detection and tracking are fast and subpixel accurate, but are not robust to large inter-frame displacements due to rotation, scale, or translation. This thesis presents an alternative hybrid approach using normalized gradient correlation (NGC) in the frequency domain and normalized cross-correlation (NCC) in the spatial domain that is fast, accurate, and robust to large displacements. A scale space search is incorporated into NGC to enable more consistent recovery of scale factors up to 6. Results show that the scale space enhanced NGC improves performance in both speed and maximum scale recovery. The proposed hybrid approach is compared to KLT and results demonstrate a significant improvement in robustness in exchange for a slight reduction in accuracy.

# Contents

# List of Figures

# List of Tables

# Acknowledgment

I would like to take this opportunity to extend my thanks to my advisor, Dr. Goshtasby, for all of the support and technical guidance he provided throughout the research and writing process. I would like to thank the members of my committee, including Dr. Goshtasby, Dr. Wischgoll, and Dr. Vasquez for all of their valuable feedback and for their candid interest and questions during my defense. I would like to acknowledge and extend my thanks to Dr. Georgios Tzimiropoulos for sharing his matlab source code of the normalized gradient correlation algorithm. I would also like to thank my colleagues, Jim Patrick, Dr. Robinson, Dr. Clouse, Dr. Abayowa, and Dr. Taylor for the countless technical discussions and advice they provided. I would like to thank my branch chief, Clare Mikula, for her persistence in encouraging me to finally get this done. Finally, I would like to thank Dr. Vasquez for mentoring me over the years and for encouraging and pushing me to grow in many dimensions, even when it required stepping beyond my comfort zone. This thesis work was funded by the Air Force Research Laboratory.

Dedicated to

my wife, Dayna, and three wonderful children, Mia, Eli, and baby Miro. Without their endless love and support I never would have survived the countless long days and nights away from home that it took to complete this thesis.

# Introduction

## 1.1 Problem & Motivation

In remote sensing applications, accurate knowledge of the alignment between consecutive image frames in aerial video is often necessary to achieve optimal performance in downstream video processing algorithms, such as video stabilization, motion detection, and object tracking. For example, motion segmentation using multi-frame differencing assumes that the background is consistent and spatially aligned across all images being differenced. Without accurate image alignment, misaligned background regions with differing intensities can degrade motion detection performance by causing false detections. If the images to be processed are produced by a stationary sensor, estimating the alignment is a trivial problem. If the images are instead produced by a non-stationary sensor, as is the case in aerial video, the presence of translation, rotation, scale, and perspective spatial deformations between images pose a more challenging problem. The term displacement is used throughout this thesis to refer to the magnitude of these spatial deformations in rotation, scale, or translation between images to be aligned.

The process of solving this problem by estimating the spatial transformation - or mapping - between two images that differ by some unknown 2D spatial deformation is known as image registration and will be described in detail in Section 2.1. There are numerous techniques available to estimate the transformation between two consecutive video frames, but among these methods there typically exists a trade-off among speed,

accuracy, and robustness to failure. One method might be fast and more accurate, but less robust, while another method might be fast and more robust, but less accurate. In addition, the accuracy and robustness of a given method is highly dependent on the quality of the video data and the operating conditions under which the video was collected.

A variety of challenging operating conditions affect image registration accuracy and robustness, which in general can be classified as sensor, scene, and environmental operating conditions. While this is not a comprehensive list, the most significant conditions affecting registration performance are briefly reviewed here. Example images containing some of these operating conditions are presented with results in Fig. 4.1. Sensor operating conditions may include large displacement transformations, differences in image sampling, illumination, and image blur. Large displacement transformations (i.e., large rotation, scale, translation, etc.) tend to be more difficult to estimate than small displacement transformations due to potentially less image overlap, different orientation and spatial sampling resolution, and ultimately less common or consistent information between images to exploit. Discrepancies in sampling resolution and orientation are a direct result of images being a quantized representation of the real world. Image blur can occur as a result of fast sensor slewing motion or optical defocus. Both defocus and motion blur result in imagery with reduced sharpness and contrast, but motion blur adds a directional component to the blur.

Scene content, such as repetitive patterns or partial to complete uniformity of background regions, may negatively impact video image registration performance. If the scene contains non-planar regions, such as variations in terrain or tall structures, parallax can pose a significant problem for successful and accurate registration. Parallax occurs when objects are located at different distances from the sensor and appear to be displaced or move relative to one another when observed from two different viewing angles; the greater the difference in distance from the sensor is, the more significant the parallax effect will be. In addition, the presence of multiple simultaneous conditions can compound the affect on performance. For example, consider an uncompressed pair of images that differ by a 2D

spatial deformation and consist entirely of a semi-uniform background. A given registration method may be able to estimate the correct transformation, assuming small displacement between images. In contrast, if the same image pair were compressed, the signal to noise ratio will be further reduced and registration may fail.

Environmental operating conditions can include bandwidth limitations, communication issues, weather (e.g., clouds, rain) and other sources of diminished clear line of sight to the ground. If image registration is performed on the ground, the video data must first be wirelessly transmitted from the platform to the ground and as a result, bandwidth limitations and communication issues can compromise the integrity of video received on the ground. Bandwidth limitations can require higher compression rates and lead to video compression artifacts, such as blockiness. Communication failures can result in additional noise and data loss that can reduce image quality. Weather, including clouds and rain, or portions of the air vehicle may partially to completely obscure direct and clear line of sight to the ground.

**The need for a video image registration method that is near real-time, subpixel accurate, and robust to failure in the presence of the above challenging operating conditions is the driving motivation behind this research.** Several approaches exist that are near real-time, have subpixel accuracy, and are robust under a subset of the mentioned operating conditions. The standard approach to video image registration is Kanade-Lucas-Tomasi (KLT) feature tracking [1]. KLT is fast, accurate, and robust to small displacement transformations. KLT can handle small amounts of noise, including video compression and data loss artifacts, and can be made partially invariant to illumination changes with a modified objective function. The primary weaknesses of KLT are large displacement transformations, nearly uniform background regions, high levels of noise, and blur. If more than one condition exists simultaneously, the problem is often compounded resulting in an even higher likelihood of KLT failure. **Finding a solution to mitigate these challenging KLT failure cases is the primary focus of this thesis.**

The following summarizes the main contributions of this thesis. A hybrid (coarse-

fine) video image registration approach was developed to be used as an alternative or complement to existing approaches, such as KLT, in order to better handle large displacement transformations. The trade space of speed, accuracy, and robustness were investigated and performance of the proposed hybrid method was compared to KLT. The results showed both methods to be viable for video image registration depending on the conditions present in the data. Methods for coarsely estimating the transformation between two images were evaluated. Several methods and sampling strategies for recovering large scales were qualitatively compared. A scale space search was incorporated into the selected coarse method to enable more consistent recovery of up to a scale factor of 6 while achieving reduced computation time. A scale factor of 6 is only significant because most test data did not contain image pairs related by scale factors greater than 6 and anecdotal results on the few that did indicated a significant degradation in performance on scale factors larger than 6. The proposed hybrid method was implemented, multi-threaded, and optimized in C++ and execution time was evaluated.

The remainder of this thesis is organized as follows. An overview of image registration immediately follows in Section 2.1. The pyramidal Kanade-Lucas-Tomasi algorithm and the Fourier-Mellin transform are introduced in Section 2.2 and Section 2.3 respectively. Related work is discussed in Chapter 2. The proposed approach and implementation are described in Chapter 3 followed by results and analysis Chapter 4. The research, contributions, and future work are summarized in Chapter 5.

# Background

## 2.1 Image Registration

Image registration is a computational method to estimate the mapping (i.e., transformation) from 2D coordinates in one image to 2D coordinates in another image, where the two images differ by some unknown 2D spatial deformation. An example pair of images that differ by a 2D spatial deformation is shown in Fig. 2.1. The source and destination images are shown in Fig. 2.1(a) and Fig. 2.1(b), respectively. Registering the source image to the destination image produces the mapping from source image coordinates to destination image coordinates. The resulting mapping can be used to warp (i.e., resample) the source image to align with the destination image. The warped source image, called the resampled image, is shown in Fig. 2.1(c).

(a) Source image (b) Destination image (c) Resampled image

Figure 2.1: Example of source, destination, and resampled source images

### 2.1.1 2D Spatial Transformations

More formally, let the coordinate system in the source image be represented by $(x, y)$ and the corresponding transformed coordinate system in the destination image be represented by $(x', y')$. A 2D mapping from coordinates $(x, y)$ to $(x', y')$ is visualized in Fig. 2.2. The

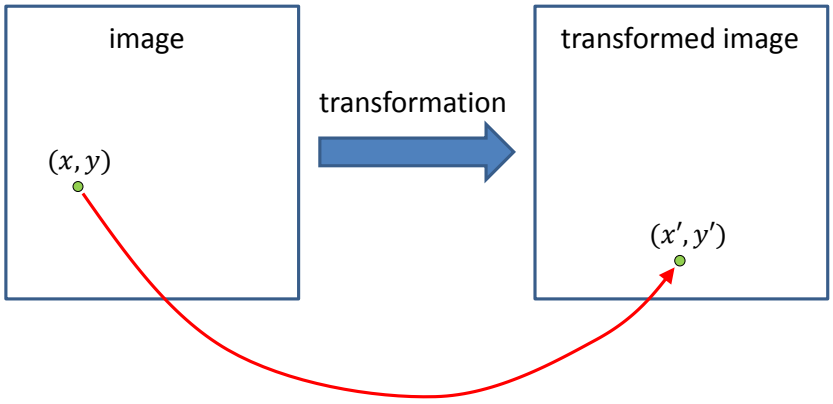



Figure 2.2: Mapping between 2D coordinates in source image and 2D coordinates in transformed image

2D mapping can be either linear or nonlinear, but nonlinear transformations are beyond the scope of this thesis. All transformations discussed and used in this thesis are linear, which is approximately valid for image registration when the effects of optical lens distortion are negligible and the ground can be reasonably approximated by a plane. In order for a transformation to be linear, it means that $x'$ and $y'$ must be a linear combination (i.e., weighted sum) of the source coordinate $(x, y)$:

$$x' = m_{11}x + m_{12}y + m_{13}$$

$$y' = m_{21}x + m_{22}y + m_{23}$$

where $m_{11}$, $m_{12}$, $m_{21}$, and $m_{22}$ are the parameters of the transformation. Alternatively, a 2D spatial transformation can be expressed in a more general form as a 3x3 matrix:

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.1}$$

where $(x, y)$ is the coordinate in the source image, $(x', y')$ is the coordinate in the destination (transformed) image, and $m_{11}$, ..., $m_{33}$ are the elements of the 3x3 transformation matrix. To incorporate translation and generalize to all types of 2D spatial transformations of up to 8 parameters, the transformation is represented by a 3x3 matrix that utilizes homogeneous coordinates in the form of a 3-element vector: $(x, y, 1)$. This is also the reason for the $w$ in the transformed homogeneous coordinate $(x', y', w)$ in Eq. (2.1). Homogeneous coordinates will be explained in greater detail in Section 2.1.4.

### 2.1.2 Similarity Transformation

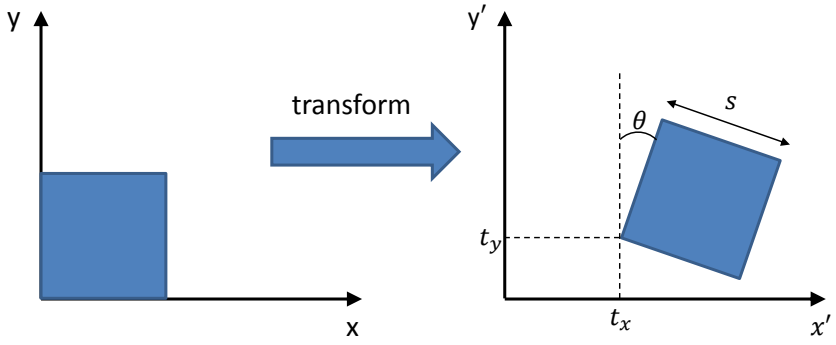



Figure 2.3: Similarity transformation

A similarity transformation is depicted in Fig. 2.3 and has 4 degrees of freedom, rotation $\theta$, uniform scaling $s$, and translation $t_x, t_y$. A similarity transformation preserves angles, parallel lines, and straight lines. Written in matrix form, a similarity transformation

is of the following form:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\theta & s\sin\theta & t_x \\ -s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where $t_x$ and $t_y$ are translation in $x$ and $y$ respectively, $\theta$ is rotation, and $s$ is uniform scaling.

### 2.1.3 Projective Transformation

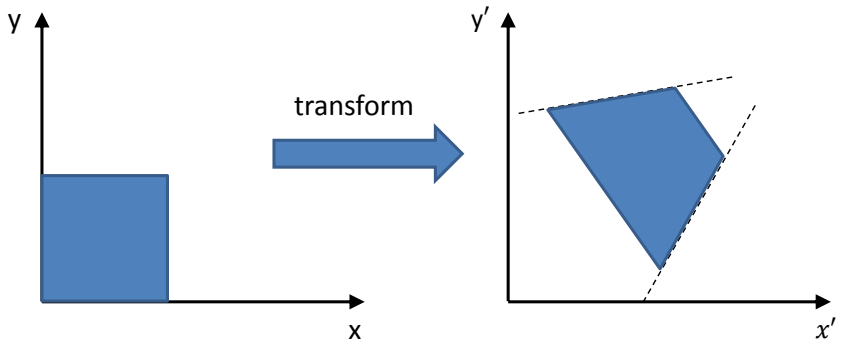



Figure 2.4: Projective transformation

A projective transformation - or homography - is depicted in Fig. 2.4 and has 8 degrees of freedom. A projective transformation preserves only straight lines. Written in matrix form, a projective transformation is of the following form:

$$\begin{bmatrix} x'w \\ y'w \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.2}$$

where $h_{11}$, ..., $h_{32}$ denote the 8 parameters, $w$ is the homogeneous scale factor, and $h_{33} = 1$.

### 2.1.4 Homogeneous Coordinates

In image registration, homogeneous coordinates serve three purposes. First, they allow 2D rotation, scale, and translation to be represented in a single 3x3 matrix. Second, they allow composition of transformation matrices, such as composing a rotation and a scaling by multiplying two transformation matrices together. Third, they enable a homography transformation matrix to manipulate 2D coordinates in 3D space (e.g., out of plane rotation or perspective). In general, homogeneous coordinates allow a transformation matrix to manipulate n-dimensional coordinate vectors in an n+1-dimensional space.

A 2D homogeneous coordinate is a column vector of the form $[x, y, 1]^T$. After being pre-multiplied by a homography matrix as in Eq. (2.2), the resulting homogeneous coordinate is a column vector of the form:

$$\begin{bmatrix} x'w \\ y'w \\ w \end{bmatrix} = \begin{bmatrix} h_{11}x + h_{12}y + h_{13} \\ h_{21}x + h_{22}y + h_{23} \\ h_{31}x + h_{32}y + h_{33} \end{bmatrix}$$

In order to convert the resulting homogeneous coordinate $[x', y', w]^T$ back to homogeneous image coordinates $[x', y', 1]$, it must be normalized as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{h_{11}x+h_{12}y+h_{13}}{h_{31}x+h_{32}y+h_{33}} \\ \frac{h_{21}x+h_{22}y+h_{23}}{h_{31}x+h_{32}y+h_{33}} \\ \frac{h_{31}x+h_{32}y+h_{33}}{h_{31}x+h_{32}y+h_{33}} \end{bmatrix}$$

In the case of a sensor that adheres to a pinhole camera model, this is known as perspective division or foreshortening. This foreshortening is what causes equidistantly spaced objects that are further from the sensor to appear closer together in image space than equidistantly spaced objects that are closer to the sensor.

### 2.1.5 Warping and Resampling

After estimating the transformation between two images, the content of one image can be spatially warped - or transformed - to align with the other image. This process involves resampling the pixel intensities in one image to spatially align with the pixel intensities in the other image. This is an important step for downstream processes that require consistent spatial alignment of multiple images. To explain how resampling works, let $I_{src}$ be the source image, $I_{dst}$ be the destination image, and $T(x, y)$ be the estimated transformation mapping from $I_{src}$ to $I_{dst}$. Then resampling $I_{src}$ to align with $I_{dst}$ is computed as follows:

$$I_{src\_warped}(x, y) = I_{src}(T^{-1}(x, y)) \tag{2.3}$$

where $(x, y)$ are coordinates in the warped source image $I_{src\_warped}$, $(x', y') = T^{-1}(x, y)$ are coordinates in original source image $I_{src}$, and $T^{-1}(x, y)$ is the inverse transformation mapping from destination coordinates to source coordinates. The resampling in Eq. (2.3) is equivalent to warping the source image $I_{src}$ to align with the destination image $I_{dst}$.

In Eq. (2.3), it is possible that the inverse transformed coordinates $(x', y') = T^{-1}(x, y)$ do not align with integral pixel offsets and as such, various interpolation methods are often employed to resample the image at subpixel offsets. The two methods used in thesis are bilinear interpolation and nearest neighbor interpolation.

#### Bilinear Interpolation

Bilinear interpolation consists of a linear weighted combination of the four surrounding pixel intensities. Let $(x_1, y_1)$, $(x_1, y_2)$, $(x_2, y_1)$, and $(x_2, y_2)$ be the four pixel coordinates surrounding the point $(x', y')$ where the interpolated image intensity $I(x', y')$ is to be computed.

Bilinear interpolation is computed by first interpolating along the $x$-direction:

$$I(x', y_1) = \frac{x_2 - x'}{x_2 - x_1} I(x_1, y_1) + \frac{x' - x_1}{x_2 - x_1} I(x_2, y_1)$$

$$I(x', y_2) = \frac{x_2 - x'}{x_2 - x_1} I(x_1, y_2) + \frac{x' - x_1}{x_2 - x_1} I(x_2, y_2)$$

Next, interpolation is performed along the $y$-direction:

$$I(x', y') = \frac{y_2 - y'}{y_2 - y_1} I(x', y_1) + \frac{y' - y_1}{y_2 - y_1} I(x', y_2) \tag{2.4}$$

**Nearest Neighbor Interpolation**

In contrast, nearest neighbor interpolation is much simpler than bilinear interpolation. In nearest neighbor interpolation, the nearest adjacent pixel coordinate to $(x', y')$ is determined. Let this nearest adjacent coordinate be denoted by $(x_{nearest}, y_{nearest})$. The image intensity at this coordinate is then used as the interpolated value:

$$I(x', y') = I(x_{nearest}, y_{nearest}) \tag{2.5}$$

Nearest neighbor interpolation is not as accurate as bilinear interpolation, but requires significantly less computation and is useful when high accuracy is not required, such as when processing binary images.

### 2.1.6 Aerial Video Image Registration

The above review of image registration provides a foundation for registering images in aerial video. The aerial video image registration problem can be viewed as a subset of the general image registration problem where images pairs are either consecutive frames or frames separated by a small temporal window from a video sequence collected by an aerial

video sensor. Solutions to aerial video image registration must be accurate, robust, and near real-time, particularly in the presence of small baseline but possibly large displacement spatial deformations. Here, baseline refers to the physical distance between the sensor location at each time an image to be registered is sensed. Displacement again refers to the magnitude of spatial change in rotation, scale, or translation between images to be registered. Even though video image registration is a small baseline problem, image pairs may still contain small but significant amounts of projective distortion.

For the video image registration problem, it can be assumed that the transformation between two consecutive images is linear due to the previously mentioned assumptions. In addition to being linear, the transformation is also assumed to be invertible, that is if $H_{ab}$ maps from image $a$ to image $b$, then $H_{ba} = H_{ab}^{-1}$ maps from image $b$ back to image $a$. Valid video frame to frame transformation matrices should be invertible. In order to verify that a 3x3 transformation matrix is invertible, the determinate must be non-zero:

$$\begin{aligned} det(M) = & \, m_{11}(m_{22}m_{33} - m_{23}m_{32}) \\ & - m_{12}(m_{21}m_{33} - m_{23}m_{31}) \\ & + m_{13}(m_{21}m_{32} - m_{22}m_{31}) \end{aligned}$$

where $M$ is the matrix in Eq. (2.1).

A method that is well suited and widely employed for video registration under ideal conditions will be discussed next.

## 2.2 Kanade-Lucas-Tomasi Feature Detection and Tracking Algorithm

The Kanade-Lucas-Tomasi (KLT) algorithm is a hybrid image registration algorithm that has long been considered the standard approach to registering consecutive frames in video due to its efficiency, accuracy, and robustness to relatively small inter-frame displacements typical in video. The term tracking comes from the fact that the KLT algorithm first detects salient features and then tracks - or locates - these features in consecutive video frames. Throughout this thesis, the term KLT is used to refer to the KLT feature detection and tracking algorithm. A brief summary of the KLT algorithm follows, but exploring KLT in greater depth is beyond the scope of this thesis and the interested reader may refer to [1]–[3] for additional details.

KLT can be summarized in three steps: detecting salient features, tracking (matching) features, and estimating a transformation from successfully tracked features. Salient corner features are first detected using *Good Features to Track* proposed by Shi-Tomasi in [3] where the saliency of a corner feature is determined using the minimum eigenvalues of the matrix $Z$ below:

$$Z = \begin{bmatrix} \sum_i g_{x_i}^2 & \sum_i g_{x_i} g_{y_i} \\ \sum_i g_{x_i} g_{y_i} & \sum_i g_{y_i}^2 \end{bmatrix}$$

where $g_{x_i}$ and $g_{y_i}$ are first order image derivatives - or gradients - in $x$ and $y$ directions respectively, summed over all pixels $i$ in a small patch encompassing the pixel of interest. Let $\lambda_1$ and $\lambda_2$ denote the eigenvalues of $Z$. If $\min(\lambda_1, \lambda_2)$ is greater than a predetermined threshold, then the corresponding pixel is detected as a good feature to track. In addition to being able to detect regions with sufficient texture in both directions, such as corners, thresholding the minimum eigenvalue also ensures that $Z$ is well conditioned for matrix inversion in the tracking process that follows.

After corners are detected in one video frame, they are located in the next consecutive

frame using an iterative image registration technique first proposed by Lucas-Kanade in [2]. The translational offset for a single image patch centered about pixel coordinate $(x, y)$ is computed using the following Newton-Raphson iterative update process:

$$\begin{aligned}
\begin{bmatrix} t_x \\ t_y \end{bmatrix}_0 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\begin{bmatrix} t_x \\ t_y \end{bmatrix}_{k+1} &= \begin{bmatrix} t_x \\ t_y \end{bmatrix}_k + \\
&\begin{bmatrix} \sum_i g_{x_i}^2 & \sum_i g_{x_i} g_{y_i} \\ \sum_i g_{x_i} g_{y_i} & \sum_i g_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i g_{x_i} [I_{dst}(x_i, y_i) - I_{src}(x_i - t_x, y_i - t_y)] \\ -\sum_i g_{y_i} [I_{dst}(x_i, y_i) - I_{src}(x_i - t_x, y_i - t_y)] \end{bmatrix}
\end{aligned} \tag{2.6}$$

where $k$ is the iteration number, $i$ is the pixel index within the image patch, $x_i$ and $y_i$ are the pixel coordinates within the image patch, $g_{x_i}$ and $g_{y_i}$ are the image gradients in the $x$ and $y$ directions respectively, and $t_x$ and $t_y$ are the tracked corner offsets from the source to destination image. The sign of the translational offsets $t_x$, $t_y$ in Eq. (2.6) is the negation of that used in [2] as the role of the source and destination images has been flipped here.

After computing $t_x$ and $t_y$ for each detected corner, the tracked corner offsets can be used to estimate the appropriate transformation parameters. This allows for a more complex transformation model to be computed from the translation only offsets of each patch. In aerial video, affine or homography are typically used depending on the level of perspective deformation present in the imagery. Methods to estimate transformation parameters from patch offsets (i.e., point correspondences) will be introduced in Section 2.4.2.

KLT by itself is only capable of estimating relatively small shifts in image patches, which results in a very small convergence domain (i.e., the maximum initial error needed to converge). The convergence domain and in turn robustness to larger displacements can be improved significantly by using the hierarchical Gaussian pyramid based approach to KLT proposed in [1]. This hierarchical approach to KLT is known as pyramidal KLT and any

usage of KLT in the remainder of this thesis refers to pyramidal KLT.

## 2.3 Fourier-Mellin Transform

This section provides an overview of the Fourier-Mellin Transform (FMT) as it pertains to image registration. The FMT is based on the Fourier-Mellin Invariant (FMI) property, which refers to the fact that the spectrum magnitude of a Fourier transformed image is invariant to translation and only variant to rotation and scaling. In contrast, the spectrum phase is variant to translation in addition to rotation and scale. This is a result of the properties of the 2D Fourier transform under rotation, uniform scale, and translation.

An image that is scaled by a factor of $a$ in the spatial domain will produce a spectrum magnitude that is inversely scaled by $1/a$ after the Fourier transform is applied. This is a result of the similarity property of the Fourier transform:

$$\mathcal{F}(f(ax, ay)) = \frac{1}{a^2} F(\frac{\omega_x}{a}, \frac{\omega_y}{a}) \tag{2.7}$$

where $x$,$y$ are spatial coordinates, and $\omega_x$,$\omega_y$ are coordinates in the frequency domain.

An image that is rotated by an angle of $\theta$ in the spatial domain will yield a spectrum magnitude that is rotated by the same angle $\theta$ in the frequency domain. This is a result of the rotation property of the Fourier transform:

$$\mathcal{F}(f(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)) = F(\omega_x\cos\theta + \omega_y\sin\theta, -\omega_x\sin\theta + \omega_y\cos\theta) \tag{2.8}$$

An image that is translated in the spatial domain will result in a spectrum phase shift in the frequency domain with no change to the spectrum magnitude. This is a result of the

Fourier shift property:

$$\mathcal{F}(f(a+t_x, y+t_y)) = e^{i2\pi(\omega_x t_x + \omega_y t_y)} F(\omega_x, \omega_y) \tag{2.9}$$

Together, the similarity, rotation, and shift properties of the Fourier transform are the motivation behind the Fourier-Mellin Transform, which is invariant to translation and variant to rotation and scale as described by the FMI above. For a deeper understanding, the derivations of the similarity, rotation, and shift properties of the Fourier transform are provided in Appendix A. More formally, the FMT is defined for a 2D image function $f(x, y)$ as the absolute value of the Fourier transformed image:

$$\text{FMT}(f(x, y)) = |\mathcal{F}(f(x, y))| \tag{2.10}$$

## 2.4 Related Image Registration Methods

This section covers related work in the area of video image registration. First, an overview of the different types of image registration methods is provided, followed by a review of methods relevant to video image registration, organized by type of method. Lastly, existing coarse registration methods that were considered as part of a hybrid method proposed in this thesis will be discussed.

### 2.4.1 Overview

Image registration methods can be divided into two broad categories: area-based and feature-based [4]. Area-based, sometimes called direct methods, directly compare intensity patterns via correlation or other similarity metrics and operate on the entire image or sub-regions of the image. Feature-based methods determine correspondences between salient features

across images and then estimate a transformation based on these correspondences. There are advantages and disadvantages to both area-based and feature-based methods that will be discussed later in the context of specific methods. A third category of hybrid methods also exists that combines elements of both area-based and feature-based methods to take advantage of their strengths while minimizing their weaknesses.

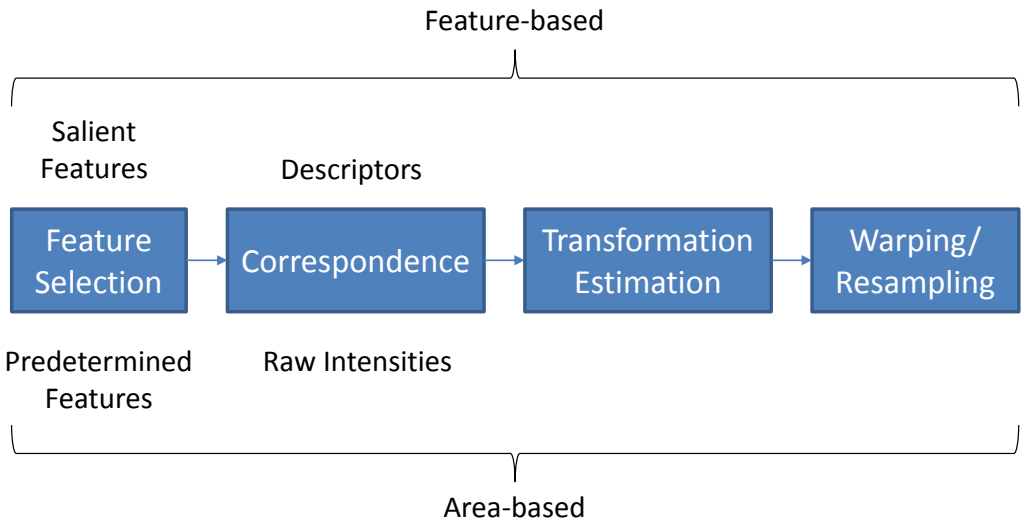



Figure 2.5: Commonality and distinction between area-based and feature-based image registration methods

To make the distinction between area-based and feature-based methods more clear, both types of methods can be viewed as a generic image registration algorithm consisting of four steps as shown in Fig. 2.5. Feature selection is the process of selecting features in the image, which could be points or regions of various size and shape up to and including the entire image. Correspondence is the process of matching features (e.g., points, regions, etc.) between the two images. Transformation estimation uses correspondence information to estimate a spatial transformation. Warping resamples the image intensities of one image to spatially align with the other image. Both feature-based and area-based methods share all four steps. The distinction lies in the first two steps. Feature-based methods detect salient features (i.e., sufficiently structured or textured) and determine correspondence by matching descriptors computed from these features whereas area-based methods use predetermined

features (e.g., regularly spaced patches, the entire image, etc.) and have no such requirement with regards to structure or texture. Feature-based methods do not typically operate directly on raw image intensities like area-based methods, however there are methods that fall into this category, which are considered to be a type of hybrid method in the context of this thesis and will be covered in detail in Section 2.4.4. In contrast, feature-based methods operate on descriptors derived from salient keypoints. Existing work is organized and reviewed based on the type of method: feature-based in Section 2.4.2, area-based in Section 2.4.3, and hybrid methods in Section 2.4.4. Lastly, a few methods are intended specifically for video stabilization and will be covered separately in Section 2.4.5.

### 2.4.2 Feature-based Methods

Following the steps in Fig. 2.5, the feature detection step produces salient keypoints, then the correspondence step provides candidate matches between feature descriptors in each image using a distance metric, and transformation estimation fits a transformation model to these correspondences, possibly with outlier removal. Research in the area of feature-based image registration often focuses on one or two of the steps in Fig. 2.5 so this logical division is used to group the related work into three areas below: feature detection, feature description and correspondence, and transformation estimation.

#### Feature Detection

There are many different approaches to detecting features - or keypoints - in images such as corners, blobs, or other distinct features that can be easily localized. An extensive and detailed overview of various types of keypoint detectors is provided by Goshtasby in [5]. When used for video image registration, a feature detector should be fast, accurate, robust to noise, repeatable, at least partially rotation and scale invariant, and able to detect distinct features.

The Harris corner detector [6] uses image gradients to approximate the curvature of the autocorrelation surface of a local image patch around each pixel to detect corners in the image. In [3], the authors modified Harris corners by performing a full eigenvalue decomposition. The Features from Accelerated Segment Test (FAST) detector [7] by Rosten and Drummond is based on the accelerated segment test, which compares the intensity of a pixel with a ring of surrounding pixels. If n out of 16 of these consecutive pixels are brighter or darker than the center pixel intensity by more than a threshold, the point is determined to be a corner. In [8], [9], Rosten et al. incorporate a machine learning approach to further improve the FAST detector. These detectors are fast, accurate, and still widely used in many applications, but lack the scale invariance necessary to handle large scale change in video.

Several feature detectors build upon some of the previous approaches. Mair et al. proposed the Adaptive and Generic Accelerated Segment Test (AGAST) detector [10] to accelerate FAST performance and removed the need for offline training through the use of binary decision trees. In [11], Leutenegger et al. contributed a multiple scale extension to AGAST designed to be used with their Binary Robust Invariant Scalable Keypoints (BRISK) descriptor. Similarly, the Oriented FAST and Rotated BRIEF (ORB) detector in [12] uses a multiple scale extension of FAST. While the multi-scale AGAST and ORB detectors are invariant to scale, their performance on large displacement transformations falls off more quickly than some of the area-based methods discussed later.

Some approaches to feature detection utilize higher order spatial information in the image, such as second order derivatives. Lindeberg [13] conducted a comparison of the trace and determinant of the Hessian matrix as a method for scale invariant blob detection. The Scale-Invariant Feature Transform (SIFT) introduced by Lowe in [14] approximates the Laplacian of Gaussian (LoG) operation with a Difference of Gaussians (DoG) and incorporates image pyramids to detect blob-like features at multiple scales. Speeded Up Robust Features (SURF) [15] uses integral images and a Hessian matrix based detector and considers multiple scales similar to SIFT. In [16], Calonder et al. compare using

FAST [7] or CenSurE [17] detectors with Binary Robust Independent Elementary Features (BRIEF) descriptors. The CenSurE detector approximates the LoG operator with a center-surround filter to reduce computation time compared to SURF/SIFT. Although there are many keypoint detectors to choose from, only the scale invariant detectors are suitable for solving the problem of large scale difference in video image registration.

### Feature Description & Matching

Feature descriptors serve as an alternative representation of an image region that attempts to capture the salient information within the region in order to facilitate accurate and efficient comparison between descriptors for applications such as image registration and object detection. For video image registration, an ideal feature descriptor should be concise (i.e., low dimensional), maintain as much salient information as possible while excluding unimportant or ambiguous information, should support computationally efficient comparison, and be invariant to certain conditions, such as scale, rotation, and illumination. Descriptors are typically stored as a vector with either floating point or binary elements. There is typically a speed accuracy trade-off between floating point and binary descriptors. Floating point descriptors tend to produce more accurate and robust results, but are significantly more computationally expensive to compute than binary features.

Only three floating point descriptors and a few variants will be mentioned here as most floating point descriptors are not capable of achieving real-time performance at video frame rates and therefore are not the primary focus of this thesis. Floating point descriptors do however provide a relevant baseline for comparison. Lowe developed the SIFT descriptor [14], which uses a Gaussian weighted histogram of oriented gradients. SIFT is one of the most well known and widely used floating point descriptors and is often used as the baseline to compare new methods against. The SURF descriptor in [15] also relies on spatial distribution of gradient information similar to SIFT, but instead utilizes a histogram of Haar wavelet responses and their absolute values. SURF also integrates gradient information

within each image subpatch. This fact, combined with the integral image Hessian-based keypoint detection mentioned above allows SURF be faster and more robust to noise than SIFT [15]. Later, Affine-SIFT (A-SIFT) [18] extended SIFT to add full local affine invariance in exchange for increased computational complexity. More recently, the KAZE descriptor [19] was shown to outperform both SURF and SIFT. Superior performance was achieved through the use of a nonlinear scale space based on nonlinear diffusion filtering that when compared to Gaussian image pyramids results in reduced noise while retaining object boundaries. SIFT was also recently extended in [20] to pool gradient orientations across different scales in addition to spatial locations to improve wide-baseline registration performance. This technique, called domain size pooling, can be applied to any other histogram based method to yield improved results on wide-baseline registration problems.

Binary descriptors are faster to compute and compare than their floating point counterparts, but they typically sacrifice accuracy and or robustness in exchange for this speed. BRIEF is a binary descriptor computed using simple intensity comparison tests for which the authors compare several different random spatial sampling distributions and levels of Gaussian smoothing [16]. BRIEF is not however invariant to rotation or scale. BRISK [11] is a binary descriptor that uses a radially symmetric sampling pattern and computes a dominant orientation to enable rotation invariance. ORB [12] is an extension of BRIEF that also computes a dominant orientation to enable rotation invariance. The dominant orientation is coarsely discretized into 12 degree increments.

Fast REtinA Keypoint (FREAK) [21] is another binary descriptor that is biologically inspired by the human retina. The FREAK descriptor uses a radially symmetric sampling pattern similar to BRISK, but samples are exponentially closer to one another near the center. The A-KAZE descriptor in [22] uses modified local difference binary matching to achieve comparable performance to KAZE in a fraction of the computational time. This modified local difference binary matching compares average gradient and intensity information over small 2x2, 3x3, 4x4, etc. patches, which is more robust than comparing single samples of

smoothed gradient or intensity. The LATCH [23] descriptor extends the idea of Three-Patch Local Binary Pattern (TPLBP) where each patch is encoded as a binary string (usually 8-bits) and compared to two surrounding patches situated on a ring of some radius and separated by some angle. The result from many of these binary comparisons is used to construct the binary descriptor. LATCH differs from the original TPLBP in that it uses more general arrangements of the three patches and learns the best arrangements based on training data.

After keypoints are detected and feature descriptors are computed, the descriptors in one image can be compared with the descriptors in another image to determine candidate matches - or correspondences. One approach is to consider the k-nearest neighbor (k-NN) candidate matches for each feature by computing a distance metric between a given feature and every other feature. The distance metric is dependent on the descriptor data type (i.e., floating point vs binary). Example distance metrics are Euclidean distance for floating point descriptors and Hamming distance for binary descriptors. In order to find the k-NN matches exactly, an exhaustive brute force search is necessary. This can be computationally prohibitive for large descriptors or large datasets so methods exist to find the approximate k-NN matches. In [24] a pyramid match kernel based approach is proposed that is capable of computing partial match similarity in time linear with the number of features. In [25], two fast approximate nearest neighbor approaches using hierarchical k-means trees and kd-trees are shown to significantly improve computational time by relaxing the requirement for optimal matching and suffering only a minimal loss to the percentage of correct neighbors.

Methods exist for efficiently discarding poor matches up front before transformation estimation. One such method is checking bidirectional consistency. If the same feature pair is found to be the best match from the source image to the destination image and conversely found to also be the best match from the destination image to the source image, the pair is considered to be a good match and if not, the pair is discarded. This requires additional computation to check matches in both directions, but can improve the quality of matches for transformation estimation. Another method is to consider the top two k-NN matches

for each candidate feature in the source image. The ratio of the distance between candidate feature and the best and second best match can be thresholded to discard matches that are two similar (i.e., ambiguous). If this ratio is high, it means the candidate feature only has one close match and little ambiguity exists. Next, methods for estimating transformation from candidate matches will be reviewed.

### Estimation of Transformation Parameters

This section describes estimation of transformation parameters from candidate correspondence pairs produced by feature detection and matching and reviews such techniques. Typically, there are more feature correspondences than needed to solve for a transformation model given the degrees of freedom in the model (i.e., the problem is overdetermined). One common approach to estimate the transformation given an overdetermined problem is least squares regression, but the least squares solution is subject to inaccuracy due to outliers - or points that are inconsistent with the model. Robust estimators are one way of dealing with these outliers. Robust estimators are capable of solving regression problems where samples are contaminated with outliers. Various robust estimators exist and a few methods commonly used for estimating transformation models from corresponding points between images are discussed here.

One such method is random sample consensus (RANSAC) [26], where a random subset of correspondences is used to estimate a transformation and any correspondences that fall outside a predefined residual error threshold are considered outliers and discarded. This process is repeated, with each iteration selecting a new random set of correspondences that satisfies a residual error threshold. The solution with the maximum number of inliers is saved and iteration continues until a maximum number of iterations is exceeded or a goodness of fit criteria is satisfied, such as the ratio of inliers to outliers meets a threshold. After outlier removal, a least squares fit is performed to compute the best fit transformation to the remaining inliers.

Due to the iterative and random nature of RANSAC, computation time can be slow and accuracy and robustness of the method is dependent on iterating enough times to randomly select a good initial subset of correspondences. Many alternatives/variants to RANSAC exist that attempt to improve computation time while maintaining accuracy and robustness. One alternative is least median of squares (LMedS) regression [27]. LMedS is an iterative process similar to RANSAC, but differs in that it computes and minimizes the median of squared residual error instead of maximizing the number of inliers that satisfy a residual error threshold. While LMedS performs well on many problems, it only works correctly when at least 50 percent of the samples are inliers.

Other well known variants of RANSAC include Preemptive RANSAC [28], PROSAC [29], and Adaptive RANSAC [30]. In Preemptive RANSAC, a fixed number of hypothesis each containing a chunk of samples is used to perform relative comparison to other hypotheses as opposed to comparing to an absolute goodness of fit metric [28]. Preemptive RANSAC is shown to outperform RANSAC when subject to a time constraint. PROSAC relies on a correspondence similarity function in order to draw samples from progressively larger sets of top-ranked correspondences compared to RANSAC where all correspondences are treated equally [29]. In order for PROSAC to out perform RANSAC, the similarity based ordering must not be worse than random ordering. Adaptive RANSAC is another type of preemptive RANSAC that reports higher accuracy and significantly less computation time than the other mentioned methods [30]. The performance of RANSAC and its variants in terms of accuracy and computation time are evaluated and compared in [30], [31].

### Performance Characteristics

Before moving on to area-based methods, the performance characteristics of feature-based methods should be considered with respect to the video image registration problem. Many of the methods reviewed above are fast, accurate, and invariant to rotation and scale. However, there are still some significant weaknesses that are relevant to the aerial video registration

problem. Based on a running time comparison of many common descriptors by Levi in [23], it can be concluded that floating-point descriptors, such as SIFT and SURF are not fast enough for near real-time video registration. Binary descriptors on the other hand are capable of near real-time performance. Feature detectors perform poorly on nearly uniform, homogeneous regions of the image. Depending on the portion and location of homogeneous regions within the image, feature-based methods may fail or incorrectly register images as a result.

Feature-based methods are sensitive to the spatial distribution of salient features in the image. Due to time constraints, feature-based registration implementations often limit the number of features considered in each image, but this can be problematic if most or all of the most salient features reside in a small region of the image. One possible solution to mitigate this problem is to reduce the sensitivity of the detector in order to detect additional features, but this results in significantly higher computation time. Another possible solution is to use a grid-based approach to force a minimum number of features to be detected in each cell of the grid. A grid-based approach achieves comparable running times, but is subject to the issue of homogeneous regions mentioned above that can contribute to poor matching performance and ultimately inaccurate or incorrect registration results. This problem occurs in aerial video and is also compounded by other sources of reduced signal-to-noise ratio, such as compression, noise, or data loss. These weaknesses motivate the consideration of area-based methods to follow.

### 2.4.3 Area-based Methods

Referring back to Fig. 2.5, area-based image registration methods do not have the explicit feature detection step that is present in feature-based methods. Instead, feature detection is implicitly accomplished by selecting one or more predetermined image regions, up to and including the entire image, to compare between images. This means that image regions are not required to contain salient features, which allows area-based methods to be more robust

to more homogeneous image regions than feature-based methods.

Another advantage is that area-based methods do not ignore or throw away as much information as compared to feature-based methods. Image regions that are lacking in salient structure can still contain valuable information for image registration, allowing area-based methods to be more robust to feature-poor imagery. By using more image information, area-based methods are capable of achieving better accuracy and robustness than feature-based methods in exchange for increased computational cost.

There are, however, some weaknesses typical to many area-based methods when compared to feature-based methods. Area-based methods that are capable of estimating sufficiently complex transformation models, such as affine or projective, tend to require greater computational time than feature-based methods. Many, but not all area-based methods, are unable to handle large displacement or wide baseline image pairs. The methods that are able to handle large displacements are typically limited to estimating a similarity transformation and thus are less accurate than feature-based methods. Given the strengths and weaknesses of both area-based and feature-based methods, an ideal solution should combine aspects of both, which provides a rationale for the hybrid approaches that will be discussed in Section 2.4.4.

There are many different measures for comparing image regions in area-based image registration. Some methods, for example, operate on raw intensities, some utilize spatial derivatives, and others apply a transform to the data, such as wavelets, Radon, or Fourier transforms. Area-based methods can be classified into two broad categories of coarse and fine based on how accurately they can estimate a homography transformation and ultimately their suitability for coarse or fine registration within a hybrid approach to video image registration. Some methods are only able to estimate similarity transformations for example and are therefor labeled as coarse methods. An overview of area-based methods is presented below based on this coarse/fine categorization.

### Coarse Methods

Wavelet transforms apply various types of wavelets (e.g., Haar wavelets) at multiple scales and are sometimes paired with cross correlation to perform image registration. An overview of several wavelet based approaches to image registration is provided in [4]. The wavelet transform preserves local frequency information over multiple scales and due to this locality is less favorable when compared to the Fourier transform for estimating global transformations with arbitrary rotation and scale. Other approaches in [32]–[34] are based on the Radon transform, which is capable of estimating rotation, uniform scale, and translation. One issue is that the Radon transform is limited to estimating scale for image deformations for which image content does not disappear outside the bounds of the image. This is problematic for finite images with large scale differences. The issue arises due to the ratio of integrals along each line over all angles in the Radon transform being used to estimate the scale. If portions of each line exist in one image, but fall outside the bounds of the other image, this ratio will produce an inaccurate scale estimate. The remainder of this section will describe more relevant area-based approaches to aerial video registration.

There are a large number of approaches based on the log polar transform (LPT) and its variants, both in the spatial and frequency domain. LPT is well known for its scale and rotation invariant properties and is well suited for estimating these two parameters for image registration. Most LPT registration methods compare a single region in each image and are limited to estimating a similarity transformation. Some of these methods are sufficiently fast and well suited for initial coarse estimates for video image registration. Some of the existing work in this area proposes hybrid techniques, but their primary contribution is in their approach to estimating rotation, uniform scale, and translation and not with respect to the refinement step in the hybrid approach they present. Due to their focus on coarse registration, these methods will be covered here as opposed to later under hybrid methods.

Log polar registration in the spatial domain followed by Levenberg-Marquardt optimization of SSD refinement is proposed in [35]. While this approach is exceptionally robust and

subpixel accurate, it is significantly slower than real-time due to an exhaustive search over all possible image regions for determining the correct log polar origin to compare against. This technique is extended in [36] to accelerate the search for the log polar origin by using a multi-resolution feature-based comparison to reduce the number of log polar resamplings and comparisons. While capable of real-time performance, the actual performance may vary based on the number of similar features that exist in the imagery. Using features in this manner is also a potential source of failure and negates some of the robustness advantage to using area-based techniques in the first place. The same approach was extended again in [37] to use Gabor features for the origin search combined with adaptive polar transform (APT) to improve upon the nonuniform sampling of the log polar transform. This approach is not real-time capable and the computation time required varies with the number of detected features similar to the previous approach. Another variant to this approach was proposed in [38], where a projective polar transform was used to reduce computational cost.

In order to estimate rotation, uniform scale, and translation, many frequency domain based approaches to image registration utilize the Fourier-Mellin Transform (FMT) discussed in Section 2.3. As previously discussed, FMT exploits the fact that scale and rotation in the spatial domain map to radially symmetric inverse scale and rotation of the spectrum magnitude in the frequency domain. In [39], conventional phase correlation is combined with FMT to estimate a similarity transformation. In [40], the use of several filters was proposed to reduce artifacts caused by the discrete finite Fourier transform.

Fitch et al. proposed orientation correlation in [41]. Orientation correlation works by correlating gradient images where the magnitude of the gradient has been normalized so that only orientation information remains. This has the advantage of treating all gradient information with equal weight regardless of the magnitude of the gradient. This helps prevent a small set of large magnitude gradients from dominating the correlation signal, such as a moving vehicle with a nearly homogeneous background around it.

Several works have attempted to address the issues associated with the nonuniform

sampling of the log polar transform, particularly the overly high density of samples near the origin with increasing sparsity moving away from the origin in the radial direction. In [42], Keller et al. proposed the Pseudopolar Fast Fourier Transform (PPFFT). The PPFFT uses a pseudopolar grid combined with the fractional Fourier transform to eliminate the need for sampling required by the log polar FFT, which results in improved performance and reduced computational complexity compared to previous approaches. In [43], Pat et al. proposed the Multilayer Fractional Fourier Transform (MLFFT), which adapts the spacing between samples with increasing levels of sampling density closer to the origin. In [44], Li et al. proposed the Multilayer Pseudopolar Fractional Fourier Transform (MLPFFT). MLPFFT is similar to MLFFT in that the sampling is progressively more dense towards the origin, but it instead uses a pseudopolar arrangement of samples.

In [45], [46], Tzimiropoulos proposed the idea of Normalized Gradient Correlation (NGC). NGC uses the complex FFT of image gradients to achieve improved robustness compared to previous log polar FFT based approaches. One major advantage is that this approach typically does not require windowing to reduce aliasing as a result of the finite FFT. In [47], Tzimiropoulos extended NGC to achieve subpixel accuracy for estimating pure translation. In [48], Kokila and Thangavel used Harris corner response as opposed to image gradients for FMT based registration.

In [49], Gonzalez proposed an another method for FMT based registration using the cepstrum. The cepstrum of an image is defined as the magnitude of the inverse Fourier transform of the logarithm of the magnitude of the Fourier transform. The performance on scale and rotation estimation was shown to be comparable to NGC. In [50], Sarvaiya et al. proposed using log-gabor filters to determine an initial scale estimate for FMT based phase correlation. The initial scale estimate was then used to downsample one of the images by a factor of 2 to enable recovery of larger scale factors.

Ren et al. proposed gradient-based subspace phase correlation in [51] capable of estimating pure translation. Results showed the 1D phase correlation used by subspace

projection to be robust to both zero mean and non-zero mean noise.

Fourier based registration methods are not capable of subpixel accuracy without upsampling or use of additional techniques so several works have proposed methods for estimating subpixel shifts. In [52], a method for estimating subpixel displacement in phase correlation is proposed that assumes the subpixel shifted image is a downsampled version of an integer shifted image. In [53], a method that uses a linear weighting between the main peak and difference between its two neighbors is proposed. This subpixel method is compared to other methods, such as fitting a Gaussian, Sinc, or Quadratic function. In [54], Guizar-Sicairos et al. proposed a method that computes the inverse FFT of phase correlation for a small high resolution region surrounding the peak. This method has equivalent accuracy to upsampling before performing the FFT on each image, but uses only a fraction of the computation time.

### Fine Methods

There is an enormous body of knowledge and research dedicated to applying, extending, and adapting optimization techniques to template tracking and image registration problems. Optimization techniques are iterative methods that attempt to maximize or minimize an objective function that measures the similarity or dissimilarity respectively between two images. Two well known similarity measures capable of cross-modal image registration are mutual information (MI) and cross cumulative residual entropy (CCRE). Two common dissimilarity measures capable of single-modal image registration are sum of squared difference (SSD) and normalized cross correlation (NCC). In [5], Goshtasby provided a detailed overview of many similarity and dissimilarity measures and evaluated their comparative performance. Optimization techniques require reasonably small error in the initial estimate of transformation parameters in order to converge. Thus optimization methods are ideally suited for small displacement registration or coarse-fine registration where a good initial estimate of transformation parameters is provided, such as from a coarser level of a multi-scale pyramid based approach or frequency domain approach to be

discussed later.

In [2], Lucas and Kanade proposed an iterative Newton-Raphson minimization of SSD that incorporates a coarse-fine approach using Gaussian image pyramids and is generalizable to affine transformations. The Lucas-Kanade approach uses what is known as an additive formulation to the optimization problem where each update to the warp parameters is performed as an incremental addition. Later, Baker and Matthews [55] proved that the additive approach is equivalent to the compositional approach where the incremental update is composed with, as opposed to added to, the warp parameters. They also showed that an inverse compositional algorithm could be derived from the forward compositional algorithm, which allows the use of more complex transformation models, such as projective warps. In this context, forward and inverse refer to exchanging the roles of the source and destination images. Baker and Matthews later presented an overview of iterative image alignment techniques [56] based on the initial work of Lucas and Kanade. In this overview they provide a unifying framework for the Lucas-Kanade algorithm and its extensions as well as examine which extensions can be used with the inverse compositional approach.

In [57], efficient second-order minimization (ESM) of SSD was proposed, which achieves second-order convergence in terms of the number of iterations and accuracy without explicitly computing the computationally expensive Hessian matrix. In [58], the inverse compositional approach to efficient second order minimization is generalized to allow for more general transformation models and shown to have equal or better convergence rates than the original inverse compositional approach.

In [59], an inverse compositional formulation for Levenberg-Marquardt optimization of MI was proposed. Results showed that with respect to SSD, optimization of MI performed well on medical images, but poorly on natural images. In [60], a method for second-order maximization of MI is proposed that maintains the wider convergence domain of gradient decent combined with the accuracy of Newton's method.

Optimization of cross cumulative residual entropy (CCRE) is used to register satellite

images in [61] and [62]. Both authors observed that CCRE had a higher convergence success rate than MI with larger initial image alignment error and that CCRE converged in fewer iterations than MI. Results also show better performance than ESM.

SSD relies on a brightness constancy constraint and is not by itself invariant to illumination changes without the addition of a photometric model. One alternative to SSD is MI or CCRE, but according to [63], MI and CCRE are capable of handling complex illumination changes at the cost of lower convergence range and high computational costs. In this sense, NCC is an attractive alternative as it is intrinsically invariant to illumination changes and has similar computational costs to SSD. In [63], optimization of NCC was extended to obtain better performance than efficient second-order optimization of SSD combined with a photometric model. NCC can also be used independent of optimization techniques to estimate translation only as in [64]. In [65], [66], Mendoza-Schrock et al. explored several methods for image registration in remote sensing applications, including a technique for grid based NCC.

Singh and Jagersand developed a modular framework in [67] that decomposes registration based trackers (i.e., optimization based registration) into three submodules: appearance model (i.e., similarity/dissimilarity measure), state space model (i.e., transformation model), and search method (i.e., optimization method). In [68], they also provide a comprehensive comparison of various combinations of optimization techniques, formulations, transformation models, and similarity/dissimilarity measures, including SSD, NCC, MI, CCRE, and many others.

### 2.4.4 Hybrid Methods

Based on the work of Kanade and Lucas in [2] and Shi and Tomasi in [3], Tomasi and Kanade proposed a hybrid coarse-to-fine approach in [1] known as Kanade-Lucas-Tomasi (KLT) tracking. As described earlier in Section 2.2, the KLT tracker combines corner detection and tracking of corners via Newton-Raphson optimization of SSD.

In [69], Wolberg and Zokai proposed a method that combines a spatial log polar coarse registration estimate with Levenberg-Marquardt minimization of SSD. This method requires significant computation time due to the exhaustive search required to determine the correct log polar origin. In [70], Crabtree et al. conducted a comparison of accuracy in translation, rotation, and scale for phase correlation, gradient correlation, normalized gradient correlation, and orientation correlation. They also presented a hybrid method combining normalized gradient correlation with an iterative technique called Fan and demonstrated improved performance.

In [71], Jackovitz developed a hybrid method for aerial video image registration and shot break detection using SURF and efficient second-order minimization of SSD. This hybrid approach is more robust thank KLT feature tracking, but it is possible to develop other hybrid approaches with even greater robustness, such the method proposed by this thesis in Chapter 3.

### 2.4.5 Video Stabilization Methods

The purpose of video stabilization is to align consecutive video frames over time, often with a temporal spatial smoothing component. This typically requires efficient video image registration as a prerequisite. In [72], three approaches to real-time keypoint tracking on mobile phones are developed and compared, including a modified SIFT tracker, modified FERNS tracker, and a patch tracker. Approaches to video stabilization in [73]–[75] all make use of KLT tracking. In [76], Yip et al. developed a framework for keypoint detection and tracking for the application of image guided surgery. Keypoint tracking performance of several methods is compared, including STAR+BRIEF, SURF, and SIFT. In [77], Veldandi et al. proposed estimating similarity transformations from aligning 1D integral projections of images. These 1D projections have the advantage of being fast and robust to noise, but spatial locality information is lost during the projection leading to lower overall robustness compared to traditional 2D correlation based approaches.

The next chapter will introduce the new method developed through this research to provide improved performance as compared to the methods presented above.

# Proposed Method

Motivated primarily by the need for a video image registration method that is robust to large displacement transformations with mild perspective differences, a hybrid coarse-fine approach was developed to overcome the weaknesses of individual feature-based and area-based methods. The speed and accuracy of individual techniques as well as combined hybrid techniques was considered to be of secondary importance. Other desirable, but lower in priority algorithm traits were considered, including robustness to image blur due to defocus or motion, semi-homogeneous background regions, and reduced image quality due to compression, data loss, haze, or partial cloud obscuration. Qualitative and quantitative comparisons of algorithms under consideration were conducted in order to design an efficient and robust hybrid approach that maintains acceptable accuracy.

**The proposed hybrid approach along with a multi-resolution scale space search and an optimized multi-threaded implementation in C++ are the primary contributions of this thesis.** Additional algorithm adaptations were made to determine registration failure or success and to improve robustness, which will be described later within the detailed steps of each algorithm. First, the trade-offs associated with choice of algorithm are discussed below. Next, the hybrid algorithm is proposed in Section 3.1, computational complexity is highlighted in Section 3.2, and lastly, implementation details are discussed in Section 3.3.

### Coarse Methods

The choice of algorithm to estimate a coarse transformation between images was largely influenced by robustness to large displacement transformations and speed. In order for a coarse technique to be useful, it must be fast enough to allow sufficient time for the refinement step that follows in a hybrid approach. Both feature-based and area-based techniques were considered with respect to their ability to recover up to a scale factor of 6, combined with arbitrary rotation and translation. The only assumption is that large scale factors will not be accompanied by large displacement translation for two images sensed within a small temporal window (i.e., a small number of frames in 30Hz video).

Numerous feature-based techniques [12], [14], [15], [21]–[23] were considered, but only the binary descriptor based techniques [12], [21]–[23] are fast enough to achieve real-time results. However, based on quantitative analysis to be reported later in Chapter 4, these binary descriptor based techniques were found to be less robust than some area-based approaches with respect to their ability to estimate large displacement transformations in aerial video. In addition to robustness to large displacement, feature-based approaches struggle to handle semi-homogeneous regions and can be negatively affected by image blur, compression, and data loss.

Some area-based techniques on the other hand are relatively more robust to semi-homogeneous regions, image blur, compression, and data loss. For the purpose of estimating a coarse transformation, techniques based on wavelets [4] and the Radon transform [32]–[34] were ruled out due to the limitations mentioned previously in Section 2.4.3. Optimization techniques [2], [55]–[58], [63], [67], [68] were also removed from consideration for coarse estimation due to limited ability to handle large displacements, even with the use of a multi-scale pyramid based approach. Lastly, techniques that utilize a log polar-like transform are robust to large displacement transformations, including arbitrary rotation, scale, and translation. Of these techniques, there are spatial domain and frequency domain approaches. The spatial domain approaches are generally not real-time due to significant computation

time required to determine the correct log polar origin for comparison. Furthermore, the process of determining the correct log polar origin is a potential point of failure. In contrast to spatial domain approaches, frequency domain approaches are capable of estimating rotation and scale without first knowing the translation. This fact combined with the computational speed of many FFT implementations make frequency domain techniques the ideal choice for real-time coarse estimation of transformations in aerial video.

Frequency domain techniques exploit the Fourier-Mellin Invariant (FMI), but differ in the type of image processed (e.g., raw intensities, gradients), the type of log polar-like sampling employed (e.g., log polar, pseudo log polar, adaptive polar), and the type of correlation computed (e.g., phase, gradient, orientation). In order to select the best performing FFT-based technique, several different techniques or variations of techniques were implemented and quantitatively compared. In addition to comparing techniques, several different configurations were tested and included in the comparison. The best performing coarse estimation technique was found to be an adaptation of normalized gradient correlation (NGC) [45] and orientation correlation (OC) [41]. The results of this comparison and analysis will be provided and discussed in Chapter 4. Various log polar-like sampling methods were qualitatively compared and anecdotal evidence showed more sophisticated sampling strategies yielded only a marginal improvement in robustness in exchange for a large increase in computational cost. **Based on this qualitative analysis, log polar sampling was chosen as the best option to use with NGC for the video image registration problem.**

### Fine Methods

Similar to the choice of coarse method, robustness and speed were the primary considerations in choosing a fine method for the hybrid approach to video image registration. Accuracy was considered next in priority after robustness and speed. Based on these criterion, robust, fast, and accurate area based-methods capable of handling small displacements were considered.

Feature-based techniques were eliminated from consideration as a registration refinement technique because detecting features is unnecessary and wasteful computation for small displacement registration. Optimization techniques [2], [55]–[58], [63], [67], [68] are another type of area-based method that fit the above criteria. A state-of-the-art optimization technique, ESM [57], was implemented and exhibited high accuracy, but robustness suffered due to convergence problems caused by the either insufficient initial error from the coarse estimate or image quality issues, such as high compression rates and near homogeneous background regions. Incorporating a multi-resolution scale space pyramid may have been one possible way to mitigate these convergence issues. **However, a grid-based normalized cross correlation technique [65], [66] was chosen instead as the fine method in favor of greater robustness at the cost of accuracy compared to optimization techniques such as ESM with later results supporting this trade-off.**

## 3.1 Hybrid Algorithm Description

Based on the above analysis, the proposed hybrid method adapts normalized gradient correlation (NGC) [45] and orientation correlation (OC) [41] for coarse estimation of a similarity transformation, followed by grid-based normalized cross correlation (NCC) to refine this similarity transformation to a full homography. The overall coarse-fine approach is depicted in Fig. 3.1 where rotation and scale (RS) are estimated independently of translation (T) using NGC and OC respectively. Pseudocode for the proposed hybrid method is provided in Appendix B.

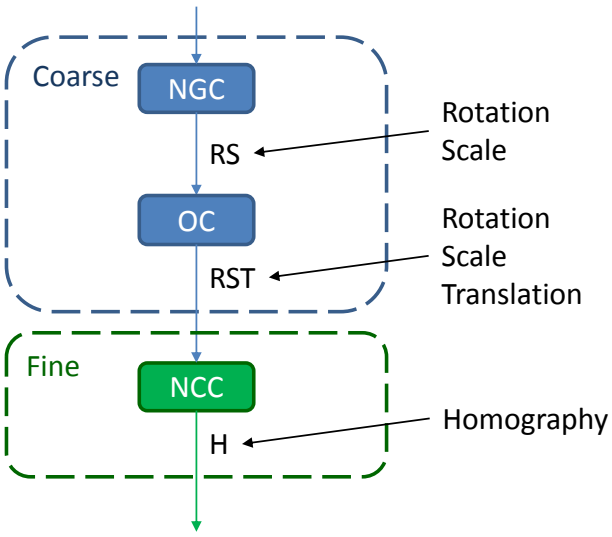



Figure 3.1: Proposed hybrid coarse-fine approach flowchart. Blue corresponds to coarse estimation of similarity parameters. Green corresponds to fine estimation of homography. The flowcharts for NGC, OC, and NCC are expanded in Fig. 3.2, Fig. 3.3, and Fig. 3.9, respectively and pseudocode is provided in Appendix B.

### 3.1.1 Coarse Algorithm

#### Normalized Gradient Correlation

Normalized gradient correlation [45] can be broken down into the six step process shown in Fig. 3.2. First, the complex gradient for each image is computed:

$$G_i = G_{i,x} + jG_{i,y} \tag{3.1}$$

where each pixel $i$ in complex gradient image $G_i$ is a complex real-valued number, $j = \sqrt{-1}$, and real component $G_{i,x} = \nabla_x I_i$ and imaginary component $G_{i,y} = \nabla_y I_i$ are the gradients in the horizontal and vertical direction respectively. Image gradients $\nabla_x I$ and $\nabla_y I$ are

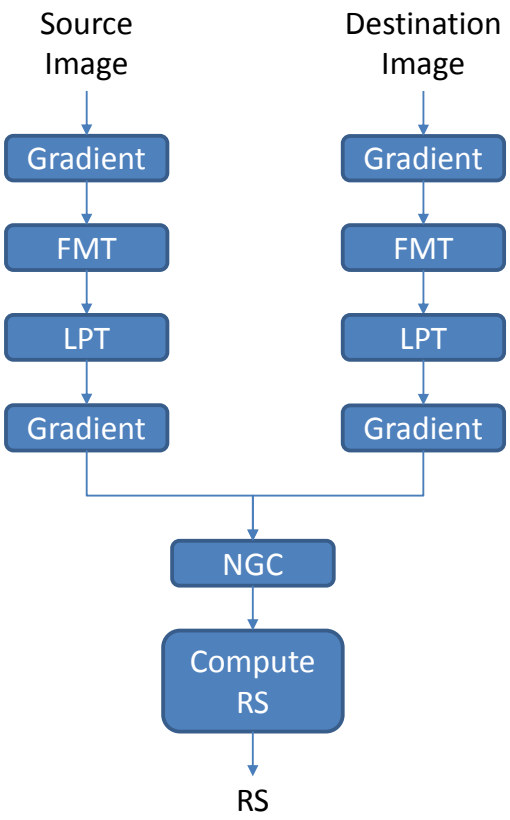



Figure 3.2: Rotation and scale estimation using NGC. NGC is the first of three sub-methods within the overall hybrid approach shown in Fig. 3.1.

computed via convolution using first-order finite central difference:

$$\nabla_x I = I * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$
$$\nabla_y I = I * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T$$

Representing the image gradients as a complex number in this way allows the gradient information to be effectively utilized by a complex FFT in the next step. When computing the FFT of each image, using the complex gradient instead of the raw intensity values significantly reduces circular wrap around aliasing caused by discontinuities near the image boundaries. The reason gradient images reduce aliasing is that high contrast edges are unlikely to occur at all or even most locations near the image boundaries in natural images so most of the pixels around the gradient image boundaries tend to be occupied by similar

small magnitude values. Some other approaches employ windowing functions (e.g., Hann, Hamming, etc.) to reduce this type of aliasing. While windowing does reduce aliasing, it has the undesired consequence of unequally weighting the frequency contribution of image content near the center compared to the edges. This can reduce the strength of the true signal in FFT-based correlation for translated and scaled images in particular. In contrast to windowing, using complex gradient images works well in practice and does not suffer from this problem.

After computing the complex gradient images, they are padded with zeros to be size $N$x$N$ where $N = 2^n$ such that $N \geq \max(\text{nrows}, \text{ncols})$ where nrows and ncols are the number of rows and columns in the complex gradient image respectively.

Second, the Fourier-Mellin transform (FMT) is applied as defined in Eq. (2.10) to compute the spectrum magnitude for each complex gradient image. The FMT allows the algorithm to take advantage of the FMI property described in Section 2.3. It holds from the FMI that rotation and scale in the spatial domain result in rotation and inverse scale respectively to the spectrum magnitude in the frequency domain, independent of translation in the spatial domain.

Third, the log polar transform (LPT) with bilinear interpolation (discussed in Section 2.1.5) is used to convert the spectrum magnitudes to log polar coordinates so that rotation and scale become pure translational offsets along the horizontal and vertical axes respectively. Only one half of the spectrum magnitude is sampled as the spectrum magnitude is periodic in the angular direction with a period of $\pi$ radians. Log polar coordinates are defined as:

$$\begin{cases} \theta = \tan^{-1}\left(\frac{y-y_c}{x-x_c}\right) \\ \rho = \log_{base}\sqrt{(x-x_c)^2 + (y-y_c)^2} \end{cases} \tag{3.2}$$

where $\theta, \rho$ are the angular and radial log polar coordinates respectively, $x, y$ are the Cartesian coordinates, and $x_c, y_c$ is the coordinate of the center pixel corresponding to the DC component of the spectrum magnitude.

Fourth, the complex gradient of the log polar transformed spectrum magnitude is computed as in Eq. (3.1). The log polar transformed spectrum magnitude is periodic along the angular direction so no discontinuity at the image boundary exists. However, discontinuity at the image boundary may exist along the scalar direction so computing the complex gradient here again has the advantage of reducing undesirable aliasing.

Fifth, normalized gradient correlation (NCC) is performed on the two complex gradients of the log polar spectrum magnitudes to recover the translational shift between the two images. This translational shift corresponds directly to rotation and scale in each direction as a result of the Fourier rotation and similarity properties defined in Eq. (2.8) and Eq. (2.7) respectively. Normalized gradient correlation [45] is defined as:

$$\text{NGC} = \frac{\mathcal{F}^{-1}(\mathcal{F}(G_{src}) \odot \mathcal{F}(G_{dst})^*)}{\mathcal{F}^{-1}(\mathcal{F}(|G_{src}|) \odot \mathcal{F}(|G_{dst}|)^*)} \tag{3.3}$$

where $G_{src}$ and $G_{dst}$ are the complex gradients from Eq. (3.1) of the source and destination images respectively, $\odot$ denotes element-wise multiplication, and $^*$ denotes the complex conjugate.

Subpixel accuracy is achieved by fitting a Gaussian peak interpolation function to the maximum peak and neighboring values of the NGC surface [78]:

$$\begin{aligned} \Delta x &= \frac{\log p(i, j-1) - \log p(i, j+1)}{2(\log p(i, j-1) - 2\log p(i, j) + \log p(i, j+1))} \\ \Delta y &= \frac{\log p(i-1, j) - \log p(i+1, j)}{2(\log p(i-1, j) - 2\log p(i, j) + \log p(i+1, j))} \end{aligned} \tag{3.4}$$

where $p$ is the normalized gradient correlation surface, $i, j$ are the row and column of the maximum peak location respectively, and $\Delta x$,$\Delta y$ are the offsets to be added to the peak

location $i, j$ to achieve subpixel accuracy. Thus the final peak location is given by:

$$\begin{aligned} peak_x &= j + \Delta\, x \\ peak_y &= i + \Delta\, y \end{aligned} \tag{3.5}$$

Finally, rotation and scale are estimated from the subpixel peak offset in Eq. (3.4) that corresponds to the largest magnitude peak in the normalized gradient correlation surface produced by Eq. (3.3):

$$\begin{aligned} \text{rotation}^\circ &= \frac{180\, peak_x}{N_\theta} \\ \text{scale} &= \exp\left(\log\left(\frac{N_s}{2}\right)\frac{2\, peak_y}{N_s}\right) \end{aligned} \tag{3.6}$$

where $x, y$ is the peak offset, $N_\theta$ is the number of samples in the angular direction of the LPT, and $N_s$ is number of samples in the scale direction of the LPT.

### Orientation Correlation

After estimating rotation and scale (RS), orientation correlation [41] is used to obtain a coarse estimate for translation (T). Orientation correlation consists of the four steps outlined in Fig. 3.3.

First, images may optionally be reduced in resolution to reduce computation time. This is accomplished by selecting images from the desired Gaussian image pyramid level with an initial scale estimate of one as described later under the scale space search method.

In order to compensate for rotation and scale, one of the two images is warped to align with the other image using the rotation and scale (RS) estimated by NGC above. Which of the two images is chosen to warp is dependent on the estimated scale, which will be discussed later. Rotation and scale compensation removes the rotation and scale difference between the two images, which leaves only translation to be estimated from the phase

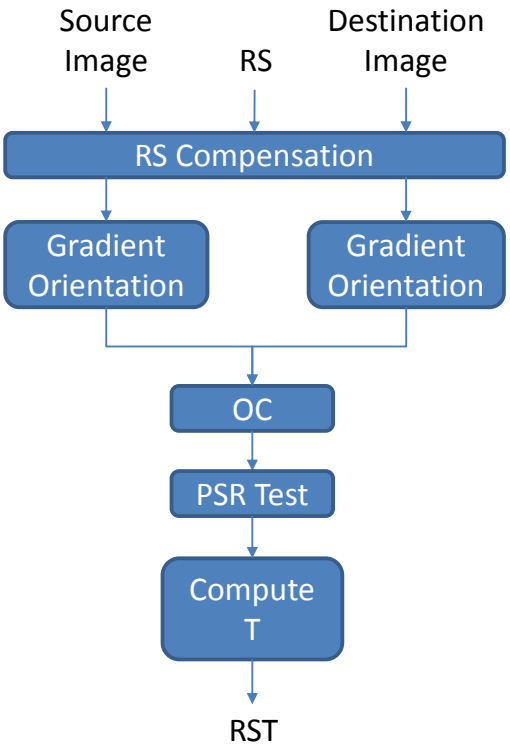



Figure 3.3: Translation estimation using OC. OC is the second of three sub-methods within the overall hybrid approach shown in Fig. 3.1.

difference of the two images Fourier spectra.

Second, the gradient orientations are computed for each RS compensated image as follows:

$$G_{i,orientation} = \begin{cases} 0 & \text{if } |G_i| < \epsilon \\ \frac{G_i}{|G_i|} & \text{otherwise} \end{cases} \tag{3.7}$$

where $G_i$ is the complex gradient defined in Eq. (3.1) and $\epsilon$ is a threshold used to zero out orientations corresponding to small magnitude gradients as they contribute to noise more so than usable signal. In order to avoid wrap around ambiguity in the next step, the complex gradient orientation images are zero padded to be the maximum of twice the size of the smaller image or the size of the larger image.

Third, by exploiting the Fourier shift property in Eq. (2.9), orientation correlation is applied to the two complex gradient orientation images to estimate the translational shift

between them:

$$\text{OC} = \mathcal{F}^{-1}(\mathcal{F}(G_{orientation,src}) \odot \mathcal{F}(G_{orientation,dst})^*) \tag{3.8}$$

where $G_{orientation,src}$ and $G_{orientation,dst}$ are the complex gradient orientations from Eq. (3.7) of the source and destination images respectively, $\odot$ denotes element-wise multiplication, and $^*$ denotes the complex conjugate.

An adaptive variant of peak-to-sidelobe ratio PSR is employed as a means for detecting registration failure. Here adaptive refers to the fact that the PSR is a relative function with respect to the mean and standard deviation and not based absolutely on the peak value. The peak and sidelobe are visualized in Fig. 3.4 where the red square is the maximum peak location and the green shaded area represents the sidelobe region. The PSR is computed by:

$$PSR = \frac{p_{max} - s_{mean}}{s_{std}} \tag{3.9}$$

where $p_{max}$ is the maximum peak value and $s_{mean}$,$s_{std}$ are the mean and standard deviation of the sidelobe region respectively. If the computed PSR is greater than a predetermined threshold, registration is considered successful, otherwise, failure occurs. This is the first of two tests for image registration failure/success. The other failure/success test occurs at the end of the fine registration algorithm.

Subpixel accuracy is again achieved by fitting a Gaussian peak interpolation function to the maximum peak and neighboring values of the OC surface:

$$\begin{aligned} peak_x &= j + \Delta\, x \\ peak_y &= i + \Delta\, y \end{aligned} \tag{3.10}$$

where $i, j$ are the row and column of the maximum peak location respectively, and $\Delta\, x$,$\Delta\, y$

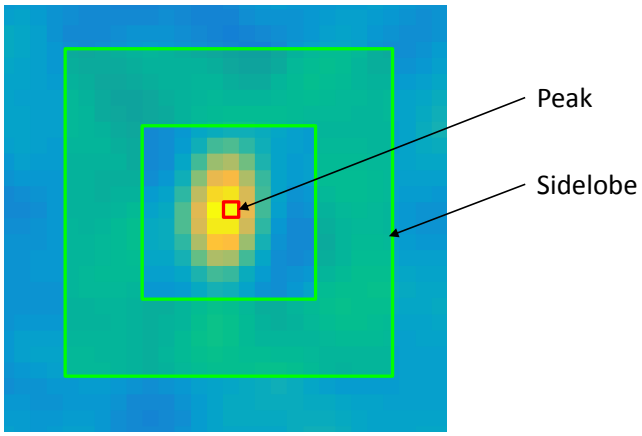



Figure 3.4: Orientation correlation surface with overlays showing regions used for PSR calculation. The red square is the maximum peak location and the green rectangular area represents the sidelobe region.

are the offsets computed by Eq. (3.4).

Finally, translation is estimated from the subpixel peak offset in Eq. (3.10) that corresponds to the largest magnitude peak in the correlation surface produced by Eq. (3.8):

$$\begin{aligned} t_x &= peak_x \\ t_y &= peak_y \end{aligned} \tag{3.11}$$

where $x, y$ is the peak offset and $t_x, t_y$ is the estimated translation in the $x$ and $y$ direction respectively.

### Scale Space Search

The coarse NGC approach outlined above can be computed quickly and efficiently by means of the FFT, however computation time is significantly slower than real-time on full resolution images from 720x480 video. In order to achieve near real-time while maintaining the ability to recover large scale factors, a multi-resolution Gaussian image pyramid based scale space search was developed to process reduced resolution imagery. A minor loss in accuracy can also be expected due to processing reduced resolution imagery, but this loss in accuracy is tolerable for the intended purpose of estimating a coarse similarity transformation.

The concept of attempting to register over multiple scales with a FMT-based approach is not new by itself. In [48], if the computed rotation and scale fail to register the two images, one image is reduced to half the resolution and the registration process is repeated. This may occur multiple times in order to search over multiple scales. This is comparable to performing a scale space search using Gaussian image pyramids as shown in Fig. 3.5 where each pair of images connected by a red arrow corresponds to one initial scale estimate to be processed by NGC. This method uses full resolution imagery as the FMT of each image requires the lower resolution image to be padded so that both images are the same dimension as shown in Fig. 3.6. The problem is that this method will perform poorly on reduced resolution images. As a solution, two methods were developed to facilitate scale space search on reduced resolution images in near real-time with minimal impact on accuracy and robustness.

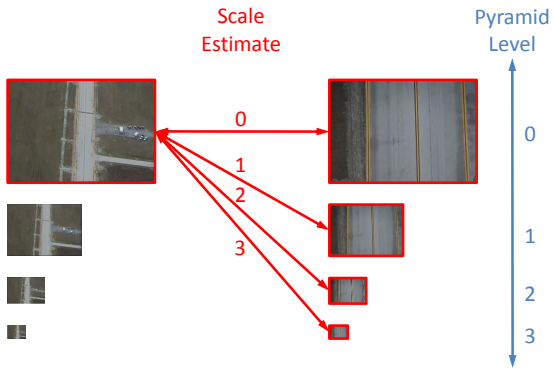



Figure 3.5: Existing multi-resolution scale space search. Non-overlapping image regions are not suppressed by cropping.

Estimating large scale factors using NGC poses two challenging issues to overcome: 1) aliasing caused by large scale differences and 2) reduced signal-to-noise ratio (SNR) due to non-overlapping image regions. These issues are compounded by the need to operate on reduced resolution imagery in order to achieve near real-time performance. The proposed scale space search methods address both of these issues by 1) reducing the relative scale difference between images to be registered and 2) applying image cropping to suppress

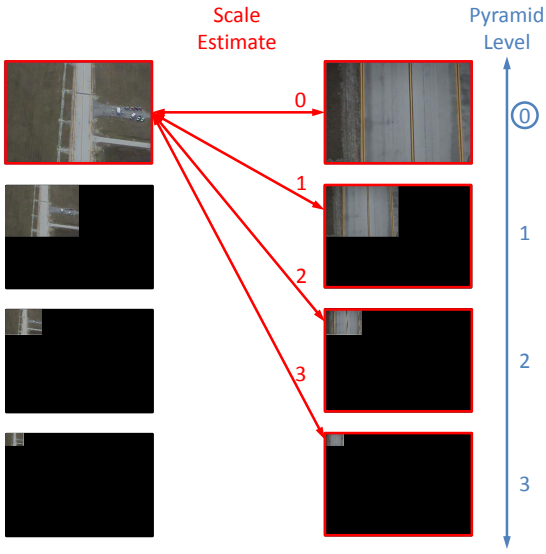



Figure 3.6: Existing multi-resolution scale space search. Non-overlapping image regions are not suppressed by cropping and padding is required as shown so that processing of original image size is necessary.

non-overlapping image regions. In both approaches, image cropping is used to suppress non-overlapping regions under the assumption that large scale factors (i.e., greater than 2) in video may only be attributed to nearly instantaneous optical zoom. Under large scale, this assumption ensures that the higher resolution image will be zoomed about or near the center of the low resolution image. This in turn ensures that the higher resolution image will not significantly overlap with the outer regions of the lower resolution image near the boundary, allowing these regions to be safely removed from consideration.

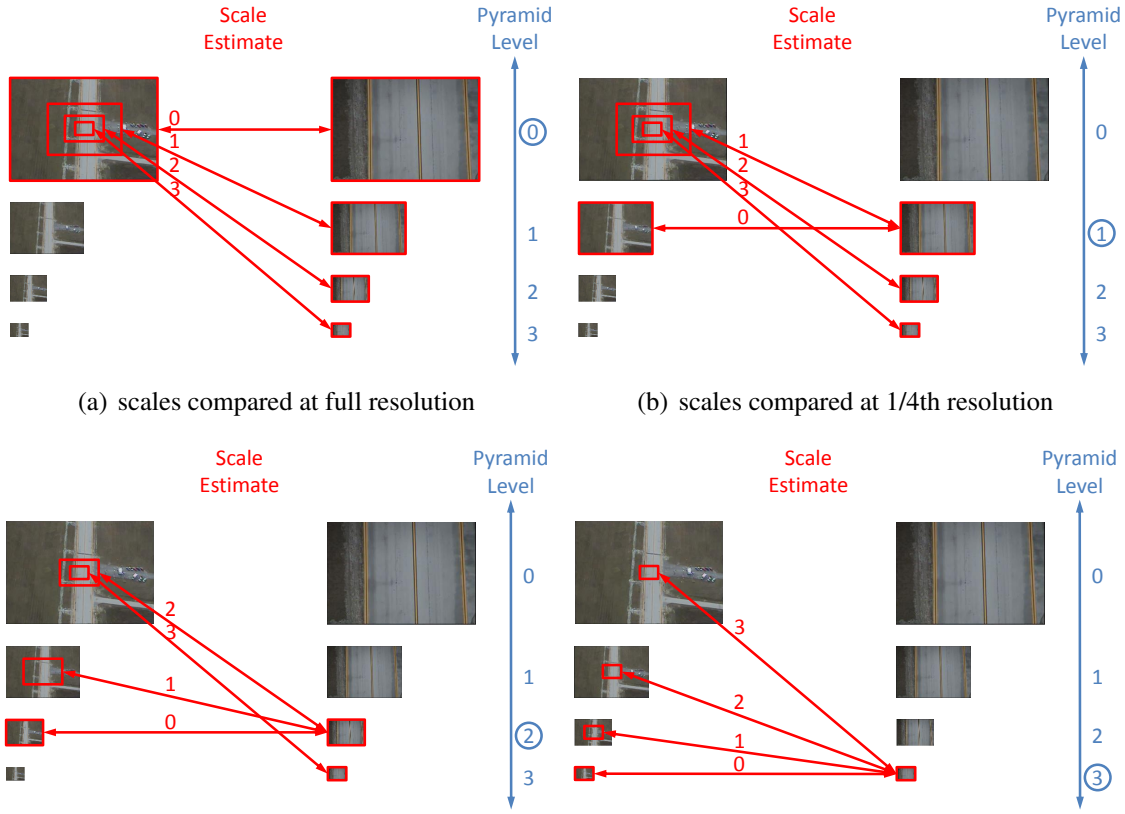
The first scale space search method, shown in Fig. 3.7, most closely resembles the existing approach, but incorporates image cropping to suppress non-overlapping regions and generalizes for processing images at any level of the pyramid. In order to process images at the resolution for a given pyramid level, each arrow is labeled with the assumed initial scale estimate and corresponds to a comparison between two image regions at the designated pyramid levels. When processing images at half resolution (i.e., pyramid level = 1) with a initial log scale estimate of 1 for example, the low resolution image region (left) at pyramid level 0 will be compared to the high resolution image (right) at pyramid level 1. For a

desired pyramid resolution level, the pyramid level and crop size used for each initial scale estimate is computed by:

$$
\begin{aligned}
n &= p_{rs} - |\log_2(s_{est})| \\
p_1 &= \max(\lceil n \rceil, 0) \\
p_2 &= p_{rs} - \min(\lceil n \rceil, 0) \\
\text{crop\_size} &= 2^{\min(n,0)-p_{rs}} * \text{image\_size}
\end{aligned}
$$

where $p_{rs}$ is the pyramid level corresponding to the desired image size to process for rotation and scale estimation, $s_{est}$ is the initial scale estimate, image_size is the input source/destination image size, crop_size is the size of the image region to crop from the left pyramid level $p_1$, and $p_1$ and $p_2$ are the left and right pyramid levels to be registered. Note that the source and destination images are swapped for scale estimates less than one.

The second scale space method is shown in Fig. 3.8 and differs from the first method in that no image region is reduced beyond the resolution of the pyramid level selected for processing. When a comparison involves two different resolution image regions, the smaller image is padded to be equal in size to the larger image. This method was selected as the proposed method over the alternate method in Fig. 3.7 on the basis of qualitative empirical evidence. It was observed that the alternate method performs better on full resolution images, but the proposed method performs better on reduced resolution images. For a desired pyramid resolution level, the pyramid level and crop size used for each initial scale




(a) scales compared at full resolution

(b) scales compared at 1/4th resolution

(c) scales compared at 1/16th resolution

(d) scales compared at 1/64th resolution

Figure 3.7: Alternate multi-resolution scale space search using Gaussian image pyramids with cropping. This approach compares pyramid levels that are as close to the estimated scale difference as possible, even if it means significantly reducing resolution. Scale estimate is the initial scale estimate assumed when comparing two images with scale values shown in red as floor$(\log_2(scale))$. Pyramid levels are shown in blue along the right-hand side.

estimate in the proposed scale search is computed by:

$$\begin{aligned} n &= p_{rs} - |\log_2(s_{est})| \\ p_1 &= \max(\lceil n \rceil, 0) \\ p_2 &= p_{rs} \\ \text{crop\_size} &= 2^{\min(n,0)-p_{rs}} * \text{image\_size} \end{aligned} \tag{3.12}$$

where $p$ is the pyramid level corresponding to the desired image size to process, $s_{est}$ is the initial scale estimate, image_size is the input source/destination image size, crop_size

is the size of the image region to crop from the left pyramid, and pyramid_level_left and pyramid_level_right are the left and right pyramid levels to be registered. Note that source and destination images are swapped for initial scale estimates less than one.

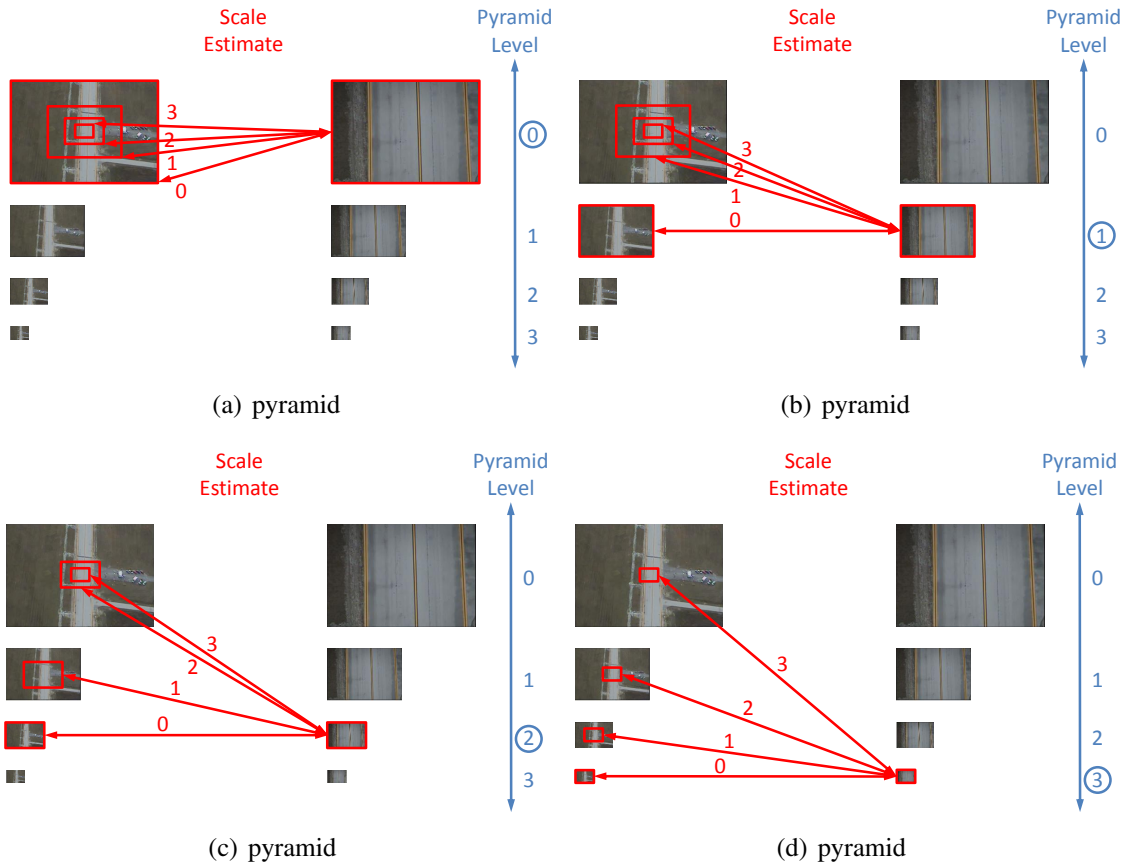



Figure 3.8: Proposed multi-resolution scale space search using Gaussian image pyramids with cropping. This approach avoids reducing resolution beyond the selected pyramid level by instead cropping one of the images to remove non-overlapping image content. Scale estimate is the initial scale estimate assumed when comparing two images with values shown in red as floor$(\log_2(scale))$. Pyramid levels are shown in blue along the right-hand side.

In order to recover large scale estimates, all possible initial scale estimates are tested and the method which yields the highest valid OC peak will be selected as the correct scale estimate. The only drawback to considering multiple scale estimates is increased computation time compared to considering only one scale estimate, but this is more than offset by the decrease in computation time as a result of processing reduced resolution images. Scale estimates of both less than one and greater than one are tested. This is

accomplished by swapping the source and destination images (i.e., left-hand and right-hand images in Fig. 3.7) in the scale space search. As previously mentioned, reducing input image resolution for translation estimation using orientation correlation can be accomplished by selecting images from the desired pyramid level with an initial scale estimate on one.

Several methods were attempted for computing a rough estimate for scale based on the Fourier spectrum magnitudes of each image. If it was possible to reliably computing a rough estimate for scale without performing NGC, it could be used to reduce the number of attempted initial scale estimates and improve computation time. One tested method applied a log-Gabor filter bank and compared the response of the filter at multiple scales. Another method compared the spreads of the one-dimensional Fourier power spectral densities. The constant term $\frac{1}{a^2}$ from Eq. (2.7) was also considered. However, none of these methods proved to be reliable in computing a coarse estimate for scale or even determining if the scale was greater than one or less than one.

### 3.1.2 Fine Algorithm

#### Grid-based Normalized Cross Correlation

After coarse registration using the adapted NGC and OC approach proposed above, the similarity transformation can be refined to a homography using a grid-based normalized cross correlation NCC technique developed by Davis and Keck at the Ohio State University and presented by Mendoza-Schrock et al. in [66]. Grid-based NCC consists of four steps as shown in Fig. 3.9. Full resolution images are processed to achieve high accuracy.

First, the lower resolution (i.e., lower zoom) image is warped to align with the higher resolution (i.e., higher zoom) image using the coarse estimation of rotation, scale, and translation (RST) and bilinear interpolation.

Second, each image is divided into a regularly spaced 2-dimensional grid. At each grid cell location, $N$x$N$ template image patches are extracted from the source image where

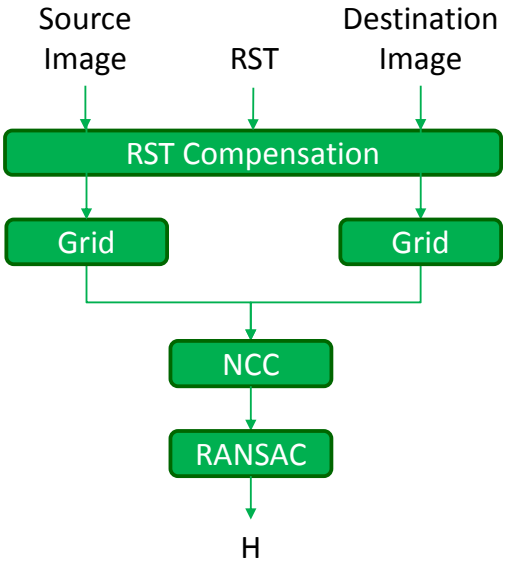



Figure 3.9: Homography estimation using grid-based NCC. NCC is the third of three sub-methods within the overall hybrid approach shown in Fig. 3.1

$N = 2n + 1$ for some integer $n > 0$. At each corresponding grid cell location in the second image, $M$x$M$ search regions are selected where $M = 2m + 1$ such that $m > n$. The value for $m$ only needs to be sufficiently large to handle the expected error in the coarse registration result. A value of $m = n + 20$ was found to be empirically sufficient for this thesis.

Third, normalized cross correlation is performed to compute the offset for each $N$x$N$ template patch in the source image within the corresponding $M$x$M$ search region in the destination image:

$$NCC(u,v) = \frac{\sum_{x,y}\left[f(x,y) - \bar{f}\right].\left[t(x-u,y-v) - \bar{t}\right]}{\sqrt{\sum_{x,y}\left[f(x,y) - \bar{f}\right]^2 \sum_{x,y}\left[t(x-u,y-v) - \bar{t}\right]^2}}$$
$$\bar{f} = \frac{1}{n}\sum_{x,y} f(x,y) \quad \bar{t} = \frac{1}{n}\sum_{x,y} t(x,y) \tag{3.13}$$

where $x, y$ are the input pixel coordinates, $u, v$ are the output correlation coordinates, $t$ is the template patch in the source image, $f$ is the search region in the destination image, $\bar{t}$ is the mean of the template region, $\bar{f}$ is the mean of the search region under the template, and

$n$ is the number of pixels in the template.

Subpixel accuracy is achieved by fitting a quadratic peak interpolation function to the peak and neighboring values of the NCC surface [79]:

$$\begin{aligned} \Delta x &= \frac{p(i,j-1) - p(i,j+1)}{2(p(i,j-1) - 2p(i,j) + p(i,j+1))} \\ \Delta y &= \frac{p(i-1,j) - p(i+1,j)}{2(p(i-1,j) - 2p(i,j) + p(i+1,j))} \end{aligned} \tag{3.14}$$

where $p$ is the normalized cross correlation surface, $i, j$ are the row and column of the maximum peak location respectively, and $\Delta x$,$\Delta y$ are the offsets to be added to the peak coordinate $i, j$ to achieve subpixel accuracy. Thus the final peak coordinate is given by:

$$\begin{aligned} peak_x &= j + \Delta x \\ peak_y &= i + \Delta y \end{aligned} \tag{3.15}$$

where $\Delta x$ and $\Delta y$ are computed using Eq. (3.14).

Finally, RANSAC is applied to correspondences obtained from the set of correlation offsets produced by Eq. (3.13) through Eq. (3.15). If a sufficient number of inlier correspondences remain after outliers are rejected by RANSAC, a homography is estimated using least squares. If the number of inlier correspondences is less than a predetermined threshold, image registration fails. The NCC estimated homography is composed with the similarity transformation estimated by the coarse approach to obtain the final homography $H$ that maps from the original source to destination image:

$$H = H_{NCC} S_{RST}$$

where $H$ is the final homography, $H_{NCC}$ is the homography computed by NCC, and $S_{RST}$ is the similarity transformation.

In summary, the hybrid method proposed above has several advantages. With the scale space search, it is capable of reliably recovering large scale factors, up to and sometimes exceeding a scale factor of 6. It is robust to large translation and rotation. Frequency domain correlation based methods such as the coarse portion of this hybrid method are robust to noise, blur, and image artifacts caused by video compression or minor data corruption or data loss. Even though the spatial domain is used, the fine portion of the hybrid method also achieves robustness to these issues by utilizing a grid-based approach combined with RANSAC. Lastly, the proposed hybrid method allows processing reduced resolution to achieve near real-time performance for recovering large scale factors of up to approximately 6 with scale space search and real-time performance for scale factors up to approximately 2 without scale space search. These values are roughly coincide with where performance starts to degrade with respect to scale with or without using scale space search, respectively.

There are also a few disadvantages to the proposed hybrid method. The coarse NGC_OC method is still prone to detecting false correlation peaks even with the employed PSR threshold test. Most of these false peaks are later ruled out by RANSAC in the fine NCC method, but these false peaks can result in more successful, but invalid registration attempts than competing methods. Another issue is aliasing. There are several ways that aliasing can negatively interfere with and prevent successful registration. This results in a small number of seemingly unexplained (to the naked eye) failure cases.

## 3.2 Computational Complexity

The computational complexity of the proposed hybrid registration algorithm as well as each sub-algorithm contained within are discussed in this section. The proposed hybrid registration algorithm consists of a coarse registration algorithm followed by a fine registration algorithm.

The coarse registration algorithm uses NGC registration to estimate rotation and scale

followed by OC registration to estimate translation. The most computationally expensive operations in NGC registration are the Fourier-Mellin transform with computational complexity of $O((n+c)^2 \log(n+c))$ and the NGC operation itself with computational complexity of $O(m^2 \log m)$, where $n$ is the image size (width=height), $c$ is the added padding discussed in Section 3.1.1 and $m$ is the log polar sampling resolution used in Eq. (3.2) with $m = N_\theta = N_\rho$.

The most computationally expensive operation in OC registration is the OC operation with computational complexity of $O((2n-1)^2 \log(2n-1))$, where $n$ denotes the image size and $2n-1$ represents the image size after padding to remove circular correlation ambiguity.

The fine registration algorithm uses grid-based NCC registration to refine the rotation, scale, and translation produced by the coarse algorithm into a full homography. In the fine algorithm, the dominant operation is normalized cross correlation, which is nested in a loop over all grid locations resulting in overall computational complexity of $O(pqt^2(w-t+1)^2)$, where $p$ and $q$ are the number of grid locations along each image axis, $w$ is the window size, and $t$ is the template size.

The above analysis of computational complexity results the following overall computational complexity for the proposed hybrid approach:

$$O(c_1 k(n+c)^2 \log(n+c) + c_2 k m^2 \log m + c_3 k(2n-1)^2 \log(2n-1) + c_4 pqt^2(w-t+1)^2)$$

which simplifies to:

$$O(c_5 k n^2 \log n + c_2 k m^2 \log m + c_4 pqt^2(w-t)^2) \tag{3.16}$$

where $c_1, c_2, c_3, c_4$ and $c_5$ are constants and $k$ is the number of tested initial scale estimates in the scale space search. Which of the three terms in Eq. (3.16) dominates the computational complexity depends on the values of constants $c_2, c_4, c_5$ and input parameters $k, m, n, p, q, t,$

and $w$. The third term grows much more rapidly than the first two terms and asymptotically dominates, but given the parameters used in practice, the computation time added by combining NGC_OC and NCC is approximately double that of NGC_OC alone as shown by the execution times presented in Table 4.5.

## 3.3 Implementation

The hybrid method is implemented in C++ and achieves near real-time when estimating scale factors greater than 2 or real-time when estimating scale factors less than 2. This is accomplished by two algorithm variations, with and without scale space search, that depending on the application, can complement or serve as an alternative to KLT. First, is the **scale space enabled**, **near real-time** hybrid method that is slightly **less accurate**, but **more robust** than KLT. This first variation is complementary to KLT as it frequently succeeds when KLT fails, but sacrifices some accuracy and as a result is better suited for recovering from occasional KLT failure. Second, is the **real-time** hybrid method **without scale space search** that is also slightly **less accurate**, but **more robust** than KLT. This second method provides an alternative to KLT when a significant increase in robustness is more important than a slight reduction in accuracy. Actual computation time for each hybrid variation will be provided and discussed in Chapter 4.

The implementation involves three levels of parallelism: cross node task, within node task, and instruction level. Flow graph, task group, and parallel for loop constructs from the Intel Thread Building Blocks[1] (TBB) is utilized for task level parallelism on the CPU. The Intel TBB flow graph is used for performing the scale space search in parallel as shown in Fig. 3.10. The scale space search corresponding to the *foreach* loop in Algorithm 1 (Appendix B) is performed in parallel via concurrent paths in the flow graph. In Fig. 3.10, this translates to three or more parallel paths where the center path handles an initial scale estimate of one, the left path(s) handle initial scale estimates greater than one, and the right path(s) handle initial scale estimates less than one. The implementation is generalized so that the set of scale estimates both greater than or less than one are configurable, hence the option to use zero or more left and right paths. Additional TBB task groups and parallel for loops are used to parallelize computations within flow graph nodes. A parallel for loop is also used in the fine registration method to perform normalized cross correlation in parallel

[1] available from https://www.threadingbuildingblocks.org/

over multiple grid locations.

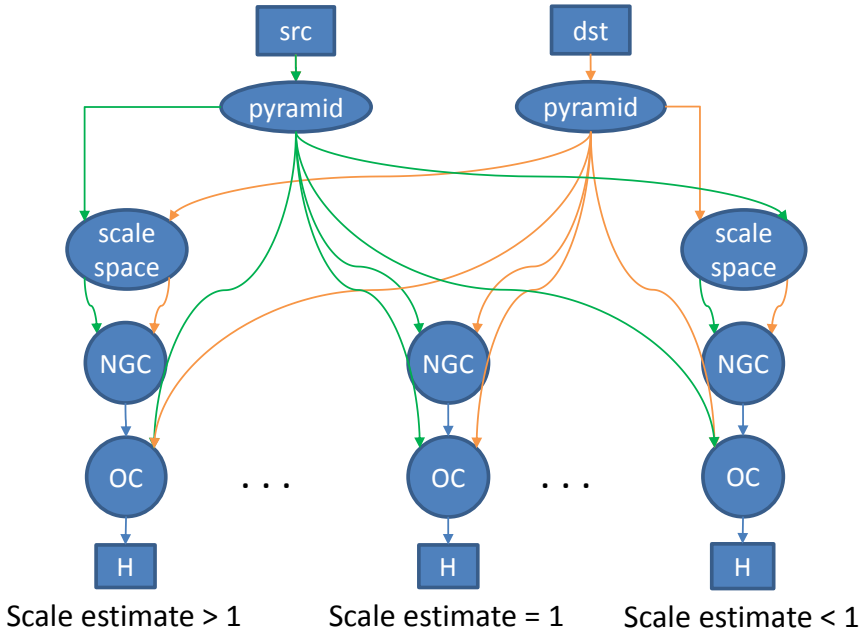



Figure 3.10: Multi-threaded implementation of scale space search using Intel TBB flow graph. Green arrows denote the source image or products derived from computations on the source image. Orange arrows denote the destination image or products derived from computations on the destination image. Blue arrows denote products that contain or are derived from both source and destination images.

Intel Streaming SIMD Extensions 2 (SSE2) are used for instruction level Single Instruction Multiple Data (SIMD) parallelism in order to compute the complex gradients and complex gradient orientations in Eq. (3.1) and Eq. (3.7), respectively. OpenCV[2] is used for image processing functions, including their highly optimized implementation of the FFT. Given the image sizes processed by the hybrid method, the OpenCV FFT implementation was found to exhibit comparable speed to other commonly used FFT implementations, such as Intel MKL and FFTW.

[2]available from http://opencv.org/

# Results

Five experiments were conducted utilizing five datasets to guide the design of the proposed coarse method and to compare the speed, accuracy, and robustness of the proposed hybrid method to other relevant methods, including KLT. The five datasets are described in Section 4.1 and will be referred to as the public dataset, restricted dataset, benchmark dataset, public Long dataset and restricted Long dataset. Note that the restricted dataset is a non-publicly releasable DoD dataset. Metrics used for evaluation are provided in Section 4.2. The five experiments are described with results presented and analyzed in Section 4.3.

## 4.1 Evaluation Datasets

The smaller datasets, public dataset, restricted dataset, and benchmark dataset, include truth data for the four corresponding corner point locations. These four pairs of corner points fully define the true homography needed to register each image pair. Truth data was included with the benchmark dataset and was hand generated to approximately pixel level or better accuracy for the public dataset and restricted dataset. An automated method for refining the hand generated truth data was avoided to prevent biasing the truth data with any one particular method. The larger datasets, public Long dataset and restricted Long dataset do not contain truth data as it was infeasible to generate by hand due to the shear volume of data. The availability of truth data impacts the evaluation metrics as described in Section 4.2.

### 4.1.1 Public Dataset

The public dataset includes 508 pairs of test images from multiple flights and two different aerial video sensors where each image in a pair is from the same flight and sensor. Sample image pairs along with the resampled image produced by the proposed hybrid registration method are shown in Fig. 4.1. Some image pairs are consecutive video frames and others are separated by less than a one second time difference. Each image pair or sequence of image pairs was hand selected specifically for the challenging operating conditions they represent. Challenging operating conditions captured in this dataset are large inter-frame displacement, including rotation, scale up to 6, and translation, motion blur, data loss artifacts, compression artifacts, nearly homogeneous background regions, mild perspective, landing gear occlusion, clouds, and parallax.

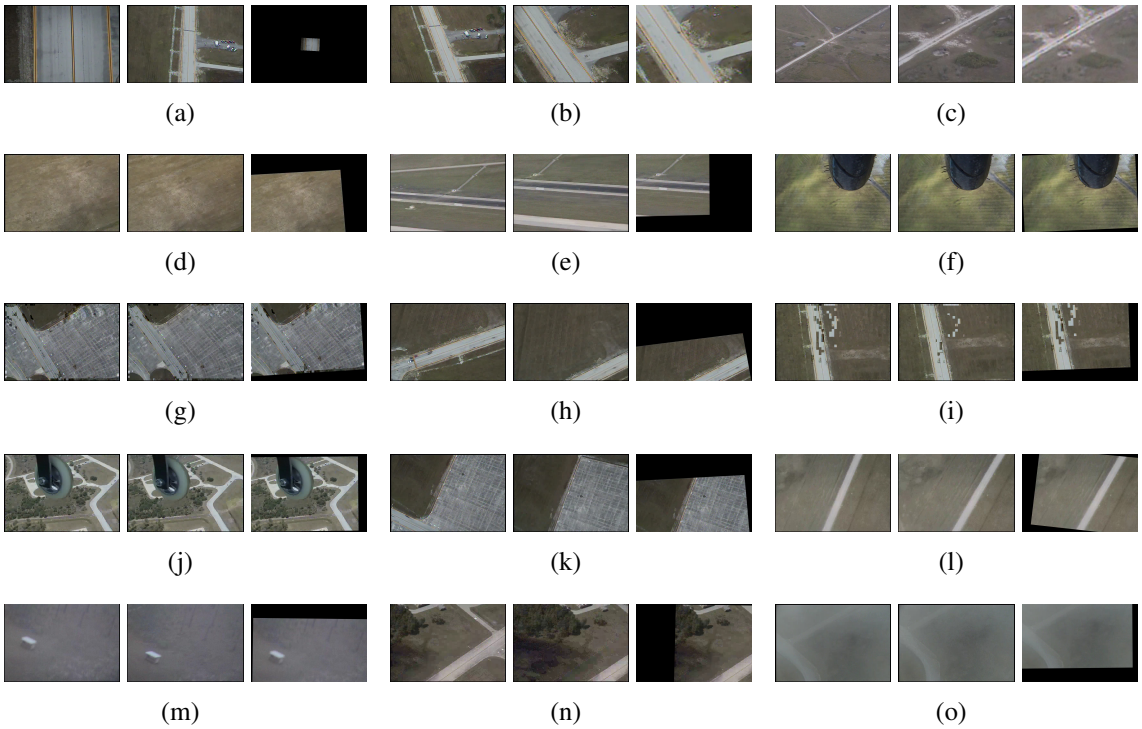



Figure 4.1: Sample image pairs from the public dataset. From left to right, each image triplet consists of source image, destination image, and resampled source image using the proposed hybrid registration method.

### 4.1.2 Restricted Dataset

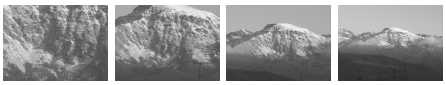
The restricted dataset consists of 52 test image pairs from two aerial video sensors that differ from above. Consistent with above, image pairs may be either consecutive or separated by a small time difference. These image pairs were chosen because either SIFT or KLT failed to register each pair. Challenging operating conditions captured in this dataset are large inter-frame displacement, including rotation, scale up to 6, and translation, compression artifacts, nearly homogeneous background regions, repeated patterns, mild perspective, and parallax. The most significant difference between this dataset and the public dataset above is that this dataset contains a large number of image pairs with scale factors near 6, whereas the public dataset only contains one image pair with a scale factor near 6.

### 4.1.3 Benchmark Dataset

The benchmark dataset[1] contains 24 test sets, totaling 295 image pairs. Sample images from 10 of these test sets are shown in Fig. 4.2. This is the standard evaluation dataset used in the literature to evaluate and compare FMT-based methods. The image pairs contain a broad range of rotation, scale, and translation values, including scale factors over 6. Large scale factors, accompanied by arbitrary rotation and small translation, is the primary challenge associated with this dataset.

[1] obtained from http://lear.inrialpes.fr/people/mikolajczyk/

(a) Bark



(b) Belledonne

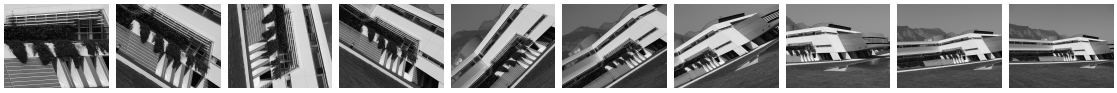
(c) Boat

(d) Crolle

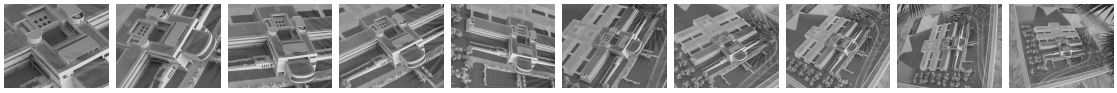
(e) East Park

(f) Sout Park

(g) Ensimag



(h) Inria



(i) Inria Model

(j) Resid

Figure 4.2: Sample images from the benchmark dataset. Registration is tested between the first (left most) image and all other images within the same test set (a) through (j).

### 4.1.4 Public Long Dataset

The public Long dataset includes a much longer, contiguous set of video frames from the same aerial video sensors as the public dataset above, with the addition of an infrared sensor. Each consecutive video frame is tested to register with the previous video frame. The videos are divided into four contiguous test sequences. The first sequence, EO Video Sequence 1, consists of 60,000 frames that contain typical inter-frame displacements. The second, EO Video Sequence 2, consists of 42,600 frames that contain a few significantly larger and more challenging inter-frame displacements than EO Video Sequence 1. The third, IR Video Sequence, consists of 20,000 frames from an infrared sensor with typical inter-frame displacements.

### 4.1.5 Restricted Long Dataset

The restricted Long dataset contains a single contiguous video sequence with 9,000 frames. This is called the Restricted Video Sequence and contains mostly typical inter-frame displacements with a few challenging image pairs having a scale factor near 6.

## 4.2 Evaluation Metrics

Different accuracy and robustness metrics were evaluated depending on the availability of truth for each test dataset. For datasets with truth, pixel location mean absolute error (MAE) is used as the accuracy metric:

$$\text{Location MAE} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i - x_{i,t})^2 + (y_i - y_{i,t})^2} \tag{4.1}$$

where $n$ is the total number of pixels in the image, $x_i, y_i$ are the pixel coordinates transformed by the estimated transformation, and $x_{i,t}, y_{i,t}$ are the pixel coordinates transformed by the

true transformation. This provides an intuitive mean euclidean error between estimated and true transformed pixel coordinates where lower values of location MAE correspond to higher accuracy and as such are more desirable.

When truth is unavailable, pixel intensity mean absolute error (MAE) is used as a measure of accuracy:

$$\text{Intensity MAE} = \frac{1}{n}\sum_{i=1}^{n} |I_{src}(T^{-1}(x_i, y_i)) - I_{dst}(x_i, y_i)| \tag{4.2}$$

where $n$ is the number of overlapping pixels, $x_i, y_i$ are the destination image coordinates, and $T^{-1}$ is the estimated transformation mapping from destination to source image. While not as informative or intuitive as Eq. (4.1), intensity MAE provides a measure that is typically, but not necessarily always, directly related to location MAE when truth is unavailable. Lower values of intensity MAE are indicative of higher accuracy and as such are more desirable.

Both accuracy metrics provide a meaningful measure of accuracy that reflects the quality of registration produced by evaluated algorithms. A good quality registration result will yield lower values of location/intensity MAE corresponding to higher accuracy and a poor quality registration result will yield higher values of location/intensity MAE corresponding to lower accuracy.

When truth is available, robustness is measured via two metrics. First, a success rate is defined as:

$$\text{success rate} = \frac{\text{\# successful and valid registration attempts}}{\text{\# total valid image pairs}} \tag{4.3}$$

where success is the success/failure status reported by the algorithm. Valid is defined as a successful registration attempt for which the estimated transformation parameters result in a location MAE that is less than a predetermined threshold. The total number of valid pairs corresponds to the number of image pairs that contain at least some human detectable

overlap as a few image pairs contain no overlap or the overlap is too difficult to detect with the naked eye. The success rate provides a measure of how often a registration algorithm produces correct result within a margin of error. Second, a false positive rate is defined as:

$$\text{false positive rate} = \frac{\text{\# of succssful, but invalid registration attempts}}{\text{\# total valid image pairs}} \tag{4.4}$$

where successful registration attempts and valid image pairs are the same as defined in Eq. (4.3) and invalid is defined as a successful registration attempt for which the estimated transformation parameters result in a location MAE that is greater than a predetermined threshold. The false positive rate provides a measure of how frequently a registration algorithm falsely reports success by producing an inaccurate or incorrect result.

When truth is unavailable, robustness can only be measured using the success/failure status reported by the algorithm as no determination of valid/invalid registration results is possible. In other words, it is not possible to determine if the registration parameters estimated by an algorithm are correct without knowing what truth is. In the case where truth is not available, the robustness metrics applied are the total number of registration attempts, the average and longest sequence of consecutive successful registration attempts, and the number of registration shot breaks. Again, success here is that reported by the algorithm, which is not necessarily consistent with truth. Shot break is defined as a contiguous span of one or more registration failures. These metrics give an idea of how often a given algorithm succeeds or fails, but no guarantee is made that the algorithm correctly determines success or failure. A small number of algorithm reported success/failure status results were validated manually, but unfortunately it is not feasible to validate all results on thousands of registered image pairs. As a result, the accuracy metric of intensity MAE is a more reliable performance metric when truth is unavailable. However, knowledge of accuracy fails to provide sufficient insight into algorithm robustness, which is the justification behind the chosen robustness metric in the absence of truth.

In summary, when truth is available, accuracy is measured by location MAE as defined in Eq. (4.1) and robustness is measured by success rate and false positive rate as defined in Eq. (4.3) and Eq. (4.4), respectively. When truth is not available, accuracy is measured by intensity MAE as defined in Eq. (4.2) and robustness is evaluated based on three metrics derived from algorithm reported results. These are the total number of registration attempts, the number of successful registration attempts, the average/longest sequence of consecutive successful registration attempts, and the number of shot breaks.

## 4.3 Experiments

### 4.3.1 Proposed and Alternate Coarse Method Evaluation

The first experiment compares the performance of the proposed coarse registration method, NGC_OC, to several alternative FMT-based coarse registration methods. Each coarse method differs only in its approach to rotation and scale estimation and a quick description of each method is provided in Table 4.1. The results of this comparison were used to select the proposed coarse method to be used in the proposed hybrid method. All coarse methods are evaluated on the public, restricted, and benchmark datasets with the success rate robustness metric defined in Eq. (4.3). The false positive rates defined in Eq. (4.4) were not compared as the responsibility of rejecting invalid coarse registration attempts is left to the fine method that follows in the hybrid approach. A MAE threshold of 15 was used to determine registration validity for this experiment. This threshold value is somewhat arbitrary, but was empirically selected as a value that provided a reasonable opportunity for the fine registration methods that follow to succeed or converge in the case of optimization methods. The accuracy metric in Eq. (4.1) is of secondary importance due to the fact that a coarse method only needs to be sufficiently accurate to serve as a rough initial estimate for successful registration of the fine method that follows. Any accuracy improvement beyond

this threshold does not necessarily contribute to improved overall accuracy of a hybrid approach.

| Method | Description |
|---|---|
| NGC_OC | normalized gradient correlation based on [45] and [41] with modifications discussed in Section 3.1.1 |
| APT_OC | combines normalized gradient correlation [45] with the adaptive polar transform [37] |
| PC_OC | traditional phase correlation [39] |
| SNGC_OC | implementation of subspace gradient correlation [51] |
| SPC_OC | subspace phase correlation inspired by [51] |
| cepstrum_OC | modified implementation of cepstrum registration [49] |
| corner_OC | implementation of corner response phase correlation [48] |

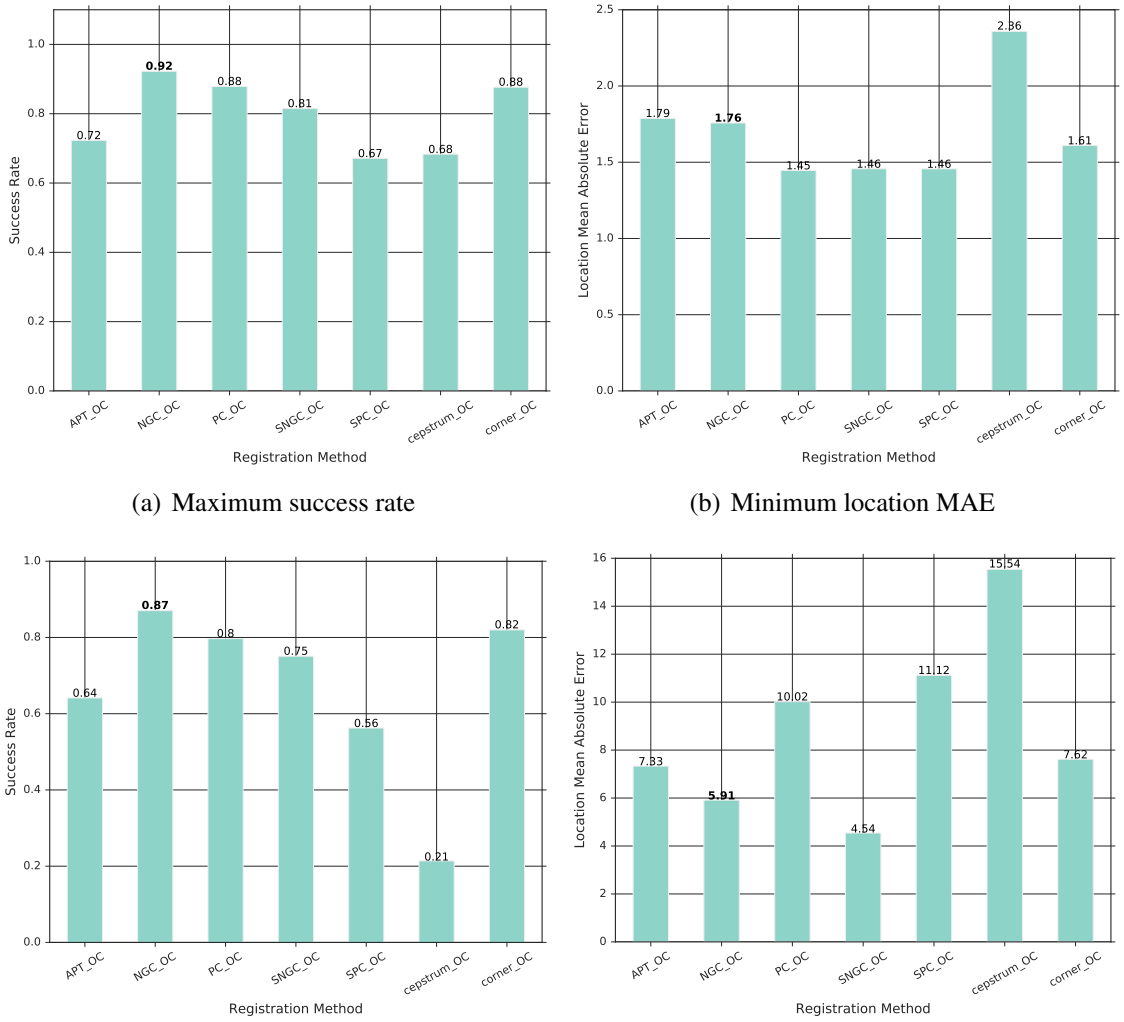
Table 4.1: Description of coarse registration methods evaluated

All methods use orientation correlation to estimate translation for two reasons. The first reason is to provide a consistent and fair comparison among FMT-based methods for rotation and scale estimation. The second is that orientation correlation gives equal weight to all frequency contributing components in the image regardless of the magnitude/contrast of the underlying structural content in the spatial domain. If the magnitude of the gradient is used for translation estimation, it can have the undesired consequence of unequally weighting small, high contrast objects, such as moving vehicles disproportionately to more homogeneous background regions. In the worst case scenario translation estimation that uses the gradient magnitude information can incorrectly register the foreground objects, such as vehicles, instead of the background. Orientation correlation discards the magnitude information and avoids this pitfall.

A parameter sweep over enabling/disabling scale space search, pyramid level, and LPT resolution was performed for each evaluated method. Two tests were conducted to compare with and without the proposed scale space search (called scale estimate in the figures below). Five combinations of pyramid levels for rotation/scale and translation estimation were tested. Three LPT sampling resolutions were tested. This totals 30 unique parameter combinations that were tested for every method on each dataset.

The overall trends in accuracy and robustness for each method on each of the three datasets is captured by the results shown in Fig. 4.3, Fig. 4.4, and Fig. 4.5. Within each figure, (a) and (b) are aggregate results over all possible parameter combinations discussed above and (c) and (d) are best results for each method. On all three datasets, NGC_OC has both the highest aggregate success rate and maximum success rate. This is a strong indication of how robust NGC_OC is compared to the other evaluated methods. Aggregate and minimum location MAE vary widely depending on method and dataset with no clear top performer, but NGC_OC exhibits acceptable location MAE for use as a coarse registration method. One possible explanation for the large variation in location MAE, even with respect to a single method, is that the formula for location MAE in Eq. (4.1) is sensitive to whether or not the estimated scale factor is greater than one or less than one, particularly for large scale factors less than one. This claim is supported by the fact that values of location MAE for all but cepstrum_OC are significantly lower on the benchmark dataset, which only contains image pairs with scale factors less than one.

The results in Fig. 4.6, Fig. 4.7, and Fig. 4.8 show aggregate success rates with respect to scale estimate for the public, restricted, and benchmark datasets, respectively. Within each figure, (a) shows the aggregate success rate with respect to scale estimate, (b) shows the aggregate success rate with respect to method and scale estimate, and (c) shows the aggregate success rate with respect to pyramid level and scale estimate for each of the three datasets. A value of true for scale estimate means that the proposed scale space search was used. Similarly, a value of false for scale estimate means that scale space search was not used. The results show that using scale space search improves success rate for all methods, at all pyramid levels, and on all datasets. The performance improvement from using scale space search is more pronounced on datasets with more test cases containing large scale




(a) Maximum success rate

(b) Minimum location MAE

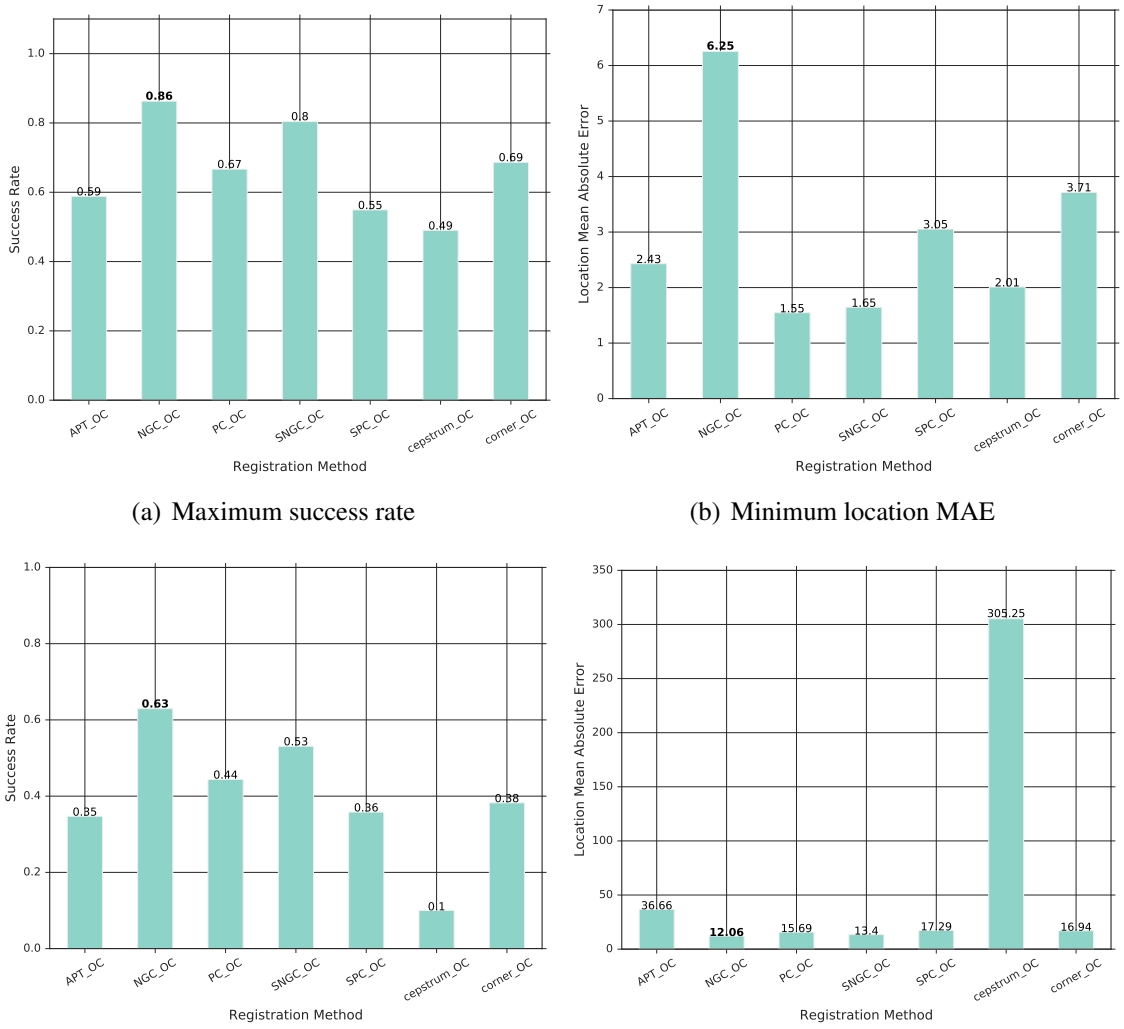
(c) Average success rate

(d) Average location MAE

Figure 4.3: Success rate and accuracy vs registration method on public dataset. (a) and (b) show the maximum success rate and minimum location MAE for each registration method over all evaluated parameter combinations. (c) and (d) show the average success rate and average location MAE aggregated over all evaluated parameter combinations.

factors, such as the restricted and benchmark datasets.

The results in Fig. 4.9, Fig. 4.10, and Fig. 4.11 again show the same data as above, but aggregated with respect to pyramid level. Within each figure, (a) shows the aggregate success rate with respect to pyramid level, (b) shows the aggregate success rate with respect to method and pyramid level, and (c) shows the aggregate success rate with respect to scale estimate and pyramid level. The displayed pyramid levels $(p_{rs}, p_t)$ correspond to the




(a) Maximum success rate

(b) Minimum location MAE

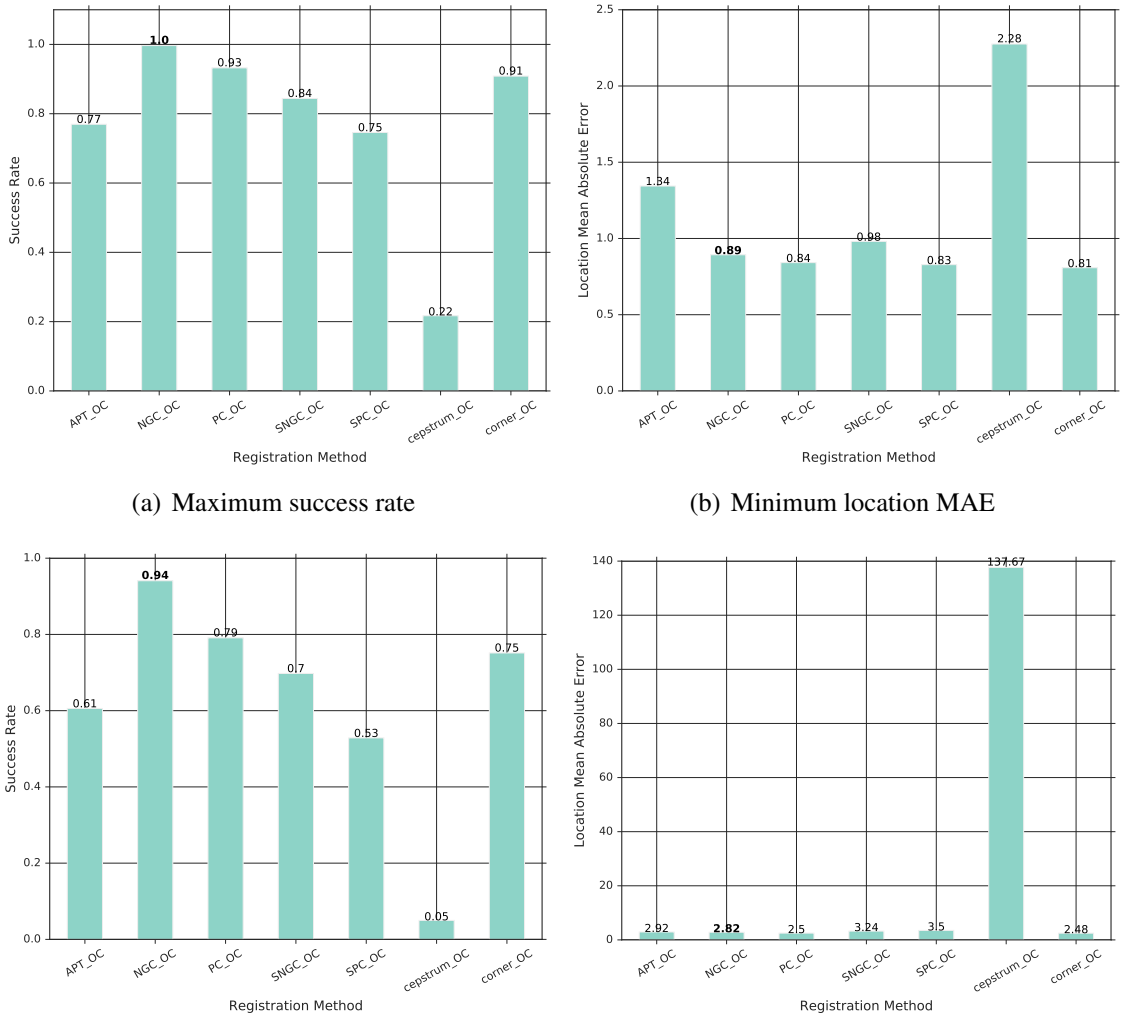
(c) Average success rate

(d) Average location MAE

Figure 4.4: Success rate and accuracy vs registration method on restricted dataset. (a) and (b) show the maximum success rate and minimum location MAE for each registration method over all evaluated parameter combinations. (c) and (d) show the average success rate and average location MAE aggregated over all evaluated parameter combinations.

pyramid level $p_{rs}$ used for rotation/scale estimation followed by the pyramid level $p_t$ used for translation estimation. A pyramid level of (1,2) for example, means that rotation and scale estimation processed images from pyramid level 1 and translation estimation processed images from pyramid level 2.

The behavior of each method with respect to pyramid level is less obvious than the previously discussed parameters. Increasing the rotation/scale pyramid level from 0 to 1




(a) Maximum success rate

(b) Minimum location MAE

(c) Average success rate

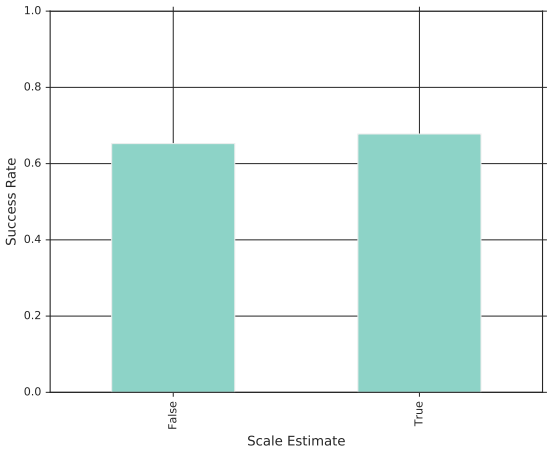
(d) Average location MAE

Figure 4.5: Success rate and accuracy vs registration method on benchmark dataset. (a) and (b) show the maximum success rate and minimum location MAE for each registration method over all evaluated parameter combinations. (c) and (d) show the average success rate and average location MAE aggregated over all evaluated parameter combinations.

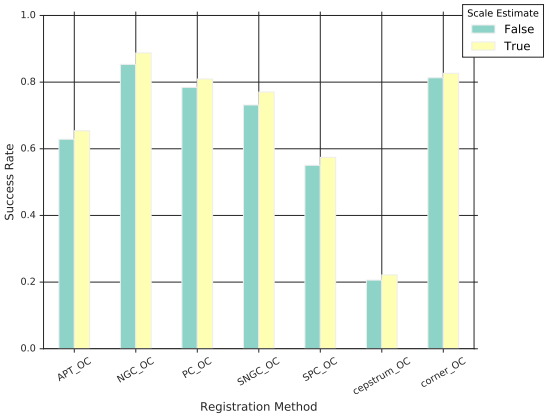
results in higher success rates on the public and restricted datasets for nearly all methods. On the other hand, the same increase in rotation/scale pyramid level on the benchmark dataset results in little change to success rate for many methods, and a slight decrease in success rate for NGC_OC in particular. A likely explanation for this behavior is that images containing more noise tend to benefit more from reducing image resolution, which in turn reduces the negative impact that the noise has on registration. This explanation is supported by the fact

that the public and restricted datasets come from video sequences using lossy compression causing distinct blocking artifacts compared to the mostly pristine imagery in the benchmark dataset. Further increasing the rotation/scale pyramid level to 2 results in a lower success rate for most methods, including NGC_OC. Considering the trade-off between speed and success rate, the best choice of pyramid level for NGC_OC is clearly 1 as level 0 requires on the order of several hundred milliseconds to execute and performance starts to suffer at level 2. Given the choice of a rotation/scale pyramid level of 1, a translation pyramid level of 2 was chosen as it provides faster performance with little to no degradation in success rate.

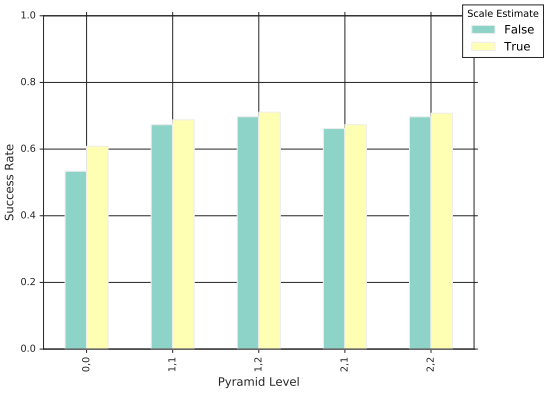
The results in Fig. 4.12 show the aggregate success rate with respect to the sampling resolution of the log polar transform. Success rates on the public, restricted, and benchmark datasets are shown in Fig. 4.12(a), Fig. 4.12(b), and Fig. 4.12(c), respectively. A log polar resolution of $N_\theta$x$N_s$ corresponds to $N_\theta$ samples in the angular direction and $N_s$ samples in the radial direction. Of the three tested log polar resolutions, the lowest resolution of 256x256 resulted in the highest aggregate success rate for all three datasets. Lower log polar sampling resolutions, such as 128x128, were also tested qualitatively, but failed to perform as well as the 256x256 resolution. There are many factors, such as sample aliasing and uneven sampling distribution, that play into which log polar sampling resolution performs the best. Without additional experimentation, it is difficult to explain exactly why one log polar sampling resolution performs better relative to another. In addition to this difficulty, the general trend does not always represent the performance of the best performing methods as will be discussed in results to follow.




(a) Average success rate vs scale estimate

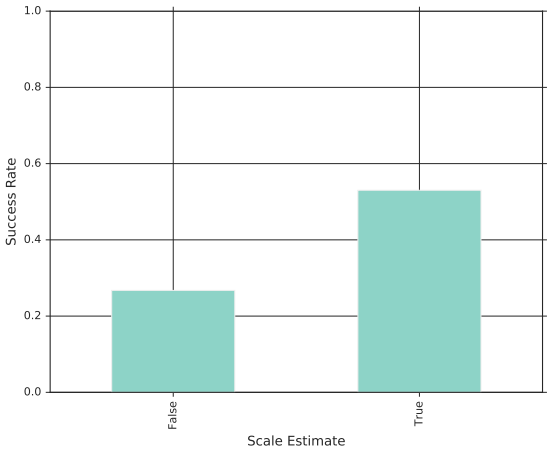



(b) Average success rate vs registration method and scale estimate

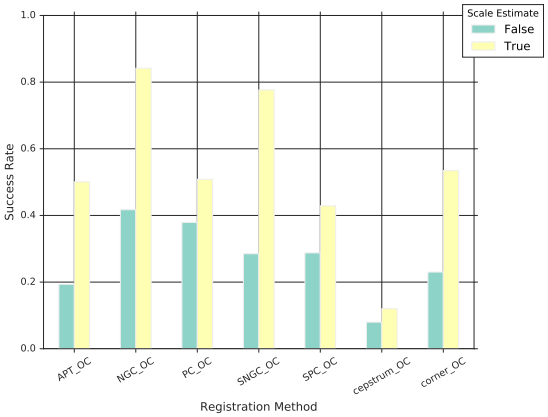



(c) Average success rate vs pyramid level and scale estimate

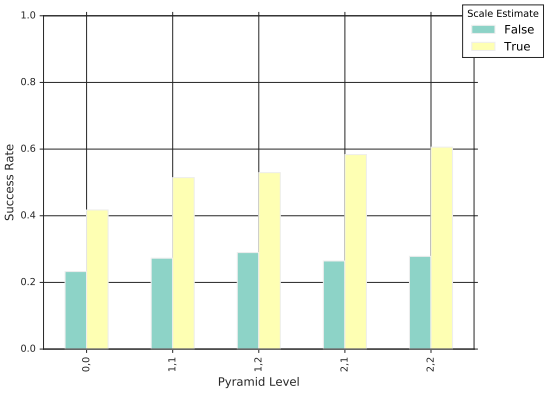
Figure 4.6: Success rate vs scale estimate on public dataset. (a) shows average success rate aggregated over registration method, pyramid level, and log polar resolution. (b) shows average success rate aggregated over pyramid level and log polar resolution. (c) shows average success rate aggregated over registration method and log polar resolution.




(a) Average success rate vs scale estimate

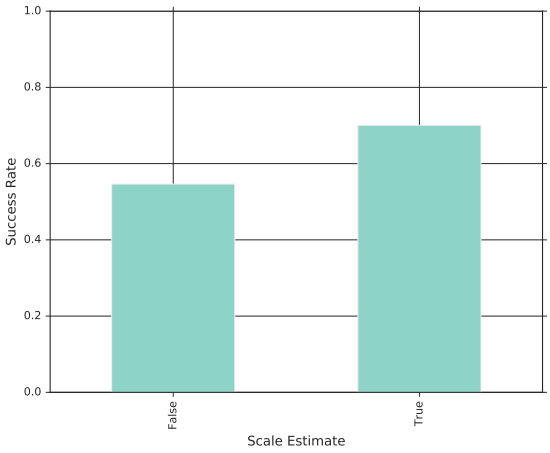



(b) Average success rate vs registration method and scale estimate

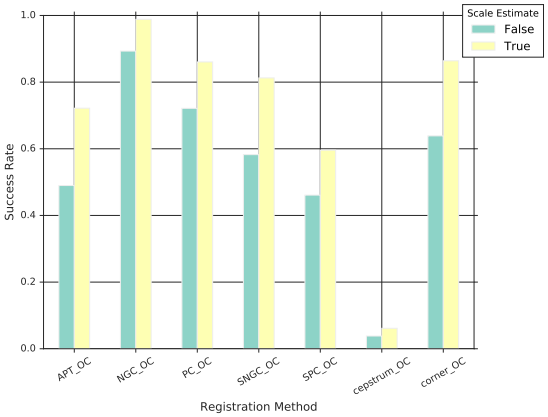



(c) Average success rate vs pyramid level and scale estimate

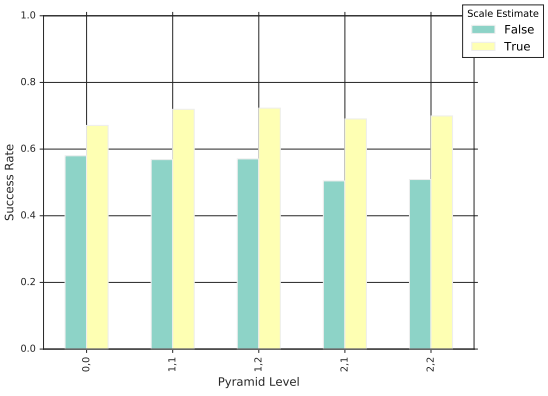
Figure 4.7: Success rate vs scale estimate on restricted dataset. (a) shows average success rate aggregated over registration method, pyramid level, and log polar resolution. (b) shows average success rate aggregated over pyramid level and log polar resolution. (c) shows average success rate aggregated over registration method and log polar resolution.




(a) Average success rate vs scale estimate

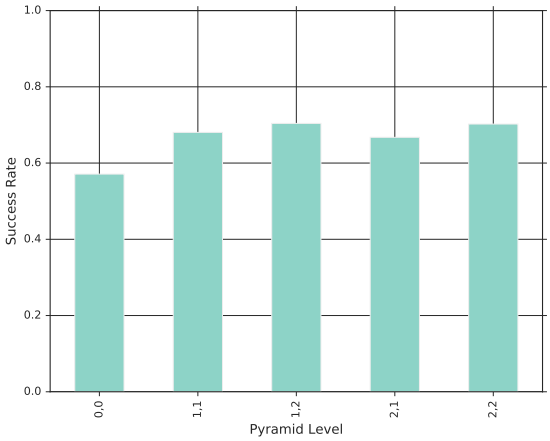



(b) Average success rate vs registration method and scale estimate

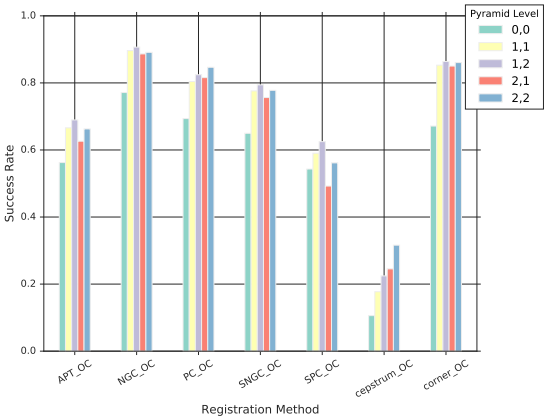



(c) Average success rate vs pyramid level and scale estimate

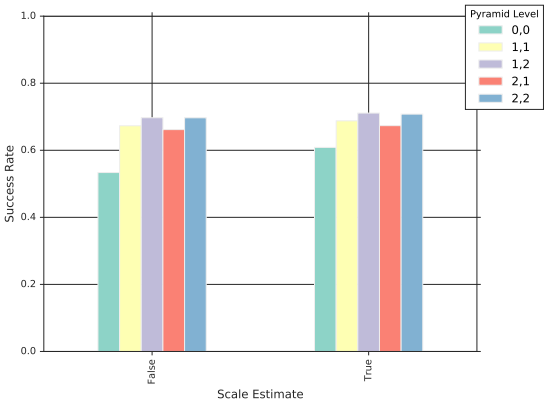
Figure 4.8: Success rate vs scale estimate on benchmark dataset. (a) shows average success rate aggregated over registration method, pyramid level, and log polar resolution. (b) shows average success rate aggregated over pyramid level and log polar resolution. (c) shows average success rate aggregated over registration method and log polar resolution.




(a) Average success rate vs pyramid level

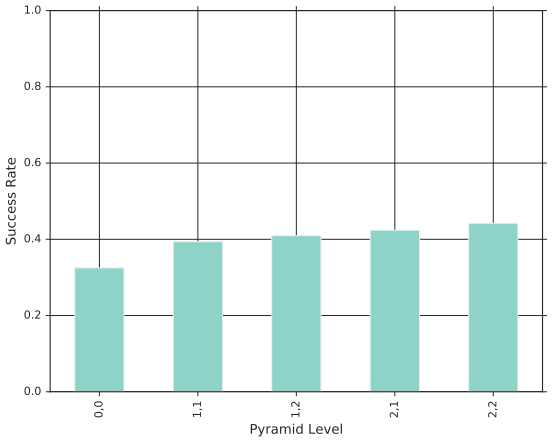



(b) Average success rate vs registration method and pyramid level

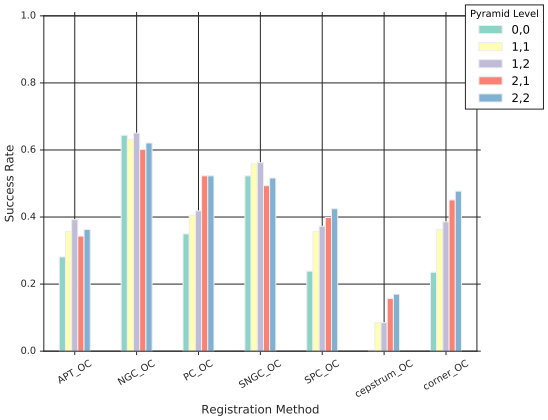



(c) Average success rate vs scale estimate and pyramid level

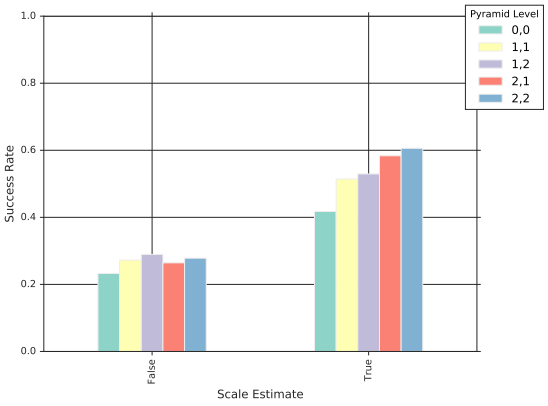
Figure 4.9: Success rate vs pyramid level on public dataset. (a) shows average success rate aggregated over registration method, scale estimate, and log polar resolution. (b) shows average success rate aggregated over scale estimate and log polar resolution. (c) shows average success rate aggregated over registration method and log polar resolution.




(a) Average success rate vs pyramid level

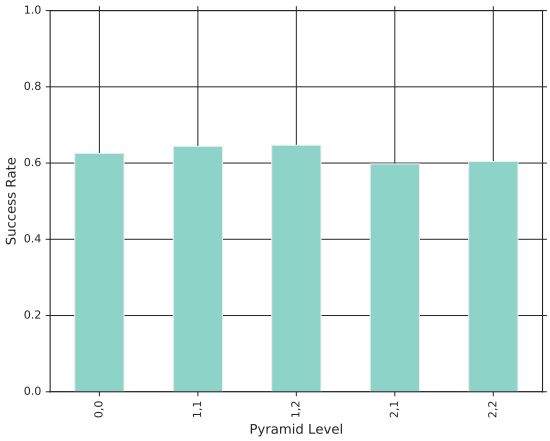



(b) Average success rate vs registration method and pyramid level

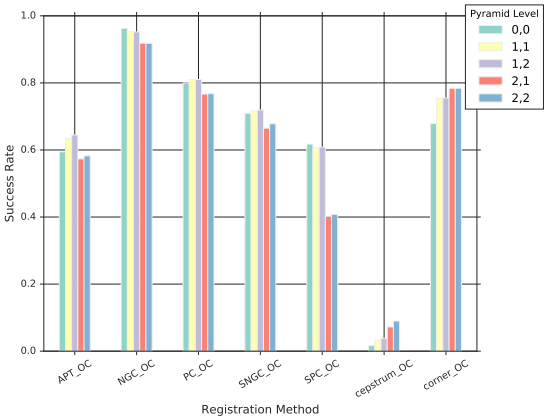



(c) Average success rate vs scale estimate and pyramid level

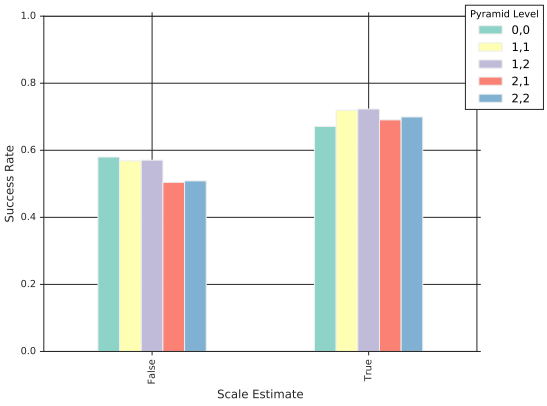
Figure 4.10: Success rate vs pyramid level on restricted dataset. (a) shows average success rate aggregated over registration method, scale estimate, and log polar resolution. (b) shows average success rate aggregated over scale estimate and log polar resolution. (c) shows average success rate aggregated over registration method and log polar resolution.




(a) Average success rate vs pyramid level

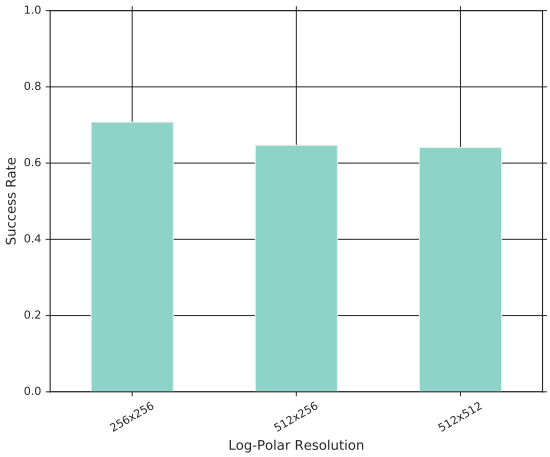



(b) Average success rate vs registration method and pyramid level

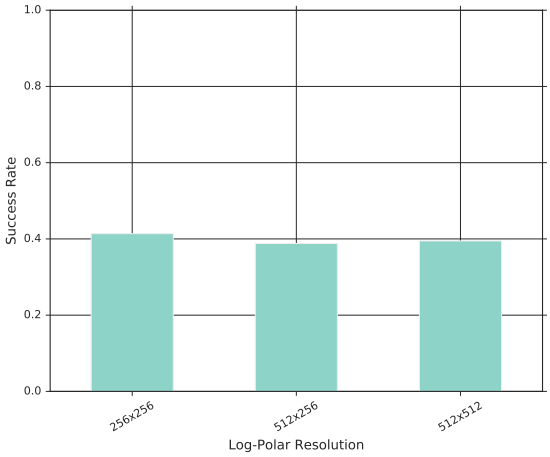



(c) Average success rate vs scale estimate and pyramid level

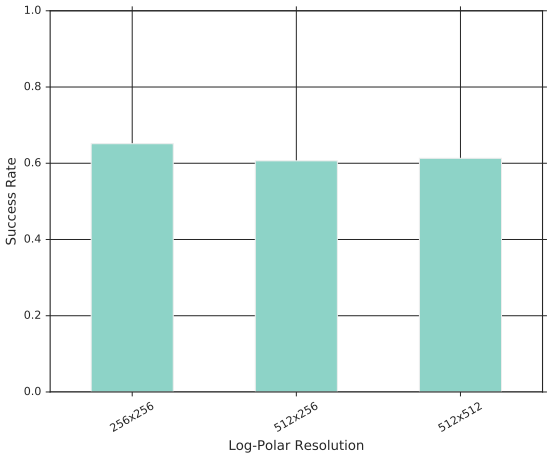
Figure 4.11: Success rate vs pyramid level on benchmark dataset. (a) shows average success rate aggregated over registration method, scale estimate, and log polar resolution. (b) shows average success rate aggregated over scale estimate and log polar resolution. (c) shows average success rate aggregated over registration method and log polar resolution.




(a) Public dataset




(b) Restricted dataset




(c) Benchmark dataset

Figure 4.12: Success rate vs log polar resolution. (a), (b), and (c) show the average success rate on the public, restricted, and benchmark datasets, respectively, aggregated over registration method, scale estimate, and pyramid level.

The previous results explain general trends observed for the evaluated methods with respect to each tested parameter. While these results are insightful, more can be learned by examining the top 10 success rates across all possible registration methods and parameter combinations on each dataset. Table 4.1(a), Table 4.1(b), and Table 4.1(c) show the top 10 success rates on the public, restricted, and benchmark datasets, respectively. Results corresponding to the selected method and parameter set are shown in bold font and confirm the choice of method and parameters discussed above. The chosen method and parameters yield the highest success rate on the public and restricted datasets and the second highest success rate on the benchmark dataset. The location MAE is not the lowest, but is competitive with other methods and acceptable considering the selected method's use for coarse registration. Notice also that even though the 256x256 log polar sampling resolution performed better in general, half or more of the top 10 parameter sets are based on other log polar sampling resolutions.

In Table 4.3, the maximum scale recovered by the proposed NGC_OC method both with and without scale space search is compared to the baseline NGC method proposed by Tzimiropoulos in [45]. Values in bold font correspond to the largest recovered scale across all methods. The NGC_OC method uses scale space search, with 256x256 log polar sampling resolution, pyramid level 1 for rotation/scale estimation, and pyramid level 2 for translation estimation. The NGC_OC (no scale) method does not use scale space search, uses 256x256 log polar sampling resolution and uses pyramid level 0 (full resolution) for rotation, scale, and translation estimation. The maximum scale recovered by proposed NGC_OC(non scale) method without scale space search is comparable to the results reported in [45] as the proposed method only differs in two aspects, the use of orientation correlation for translation estimation and peak-to-sidelobe ratio test for success/failure determination. More importantly, the results show that the proposed NGC_OC method meets or exceeds the maximum scale recovered by the baseline NGC method while using half the resolution for rotation/scale estimation and one quarter the resolution for translation estimation in a

(a) Public dataset

| Registration Method | Pyramid Level | Use Scale Estimate | Logpolar Res | Valid Truth | Avg Location MAE | Std Location MAE | Success | Valid Est | False Positive | Success Rate | False Positive Rate |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **NGC_OC** | **1,2** | **True** | **256x256** | **502** | **2.58** | **14.03** | **463** | **465** | **2** | **0.92** | **0.00** |
| | | | 512x512 | 502 | 3.16 | 19.01 | 459 | 462 | 3 | 0.91 | 0.01 |
| | | | 512x256 | 502 | 2.52 | 12.98 | 458 | 459 | 1 | 0.91 | 0.00 |
| | 2,2 | | 256x256 | 502 | 2.06 | 1.85 | 457 | 457 | 0 | 0.91 | 0.00 |
| | 1,1 | | 512x256 | 502 | 3.32 | 21.77 | 455 | 459 | 4 | 0.91 | 0.01 |
| | | | 256x256 | 502 | 1.94 | 2.20 | 454 | 456 | 2 | 0.90 | 0.00 |
| | | | 512x512 | 502 | 3.28 | 20.94 | 454 | 458 | 4 | 0.90 | 0.01 |
| | 2,2 | | 512x256 | 502 | 2.02 | 1.73 | 454 | 454 | 0 | 0.90 | 0.00 |
| | 2,1 | | | 502 | 3.46 | 21.89 | 454 | 457 | 3 | 0.90 | 0.01 |
| | 1,2 | False | 512x512 | 502 | 2.49 | 14.21 | 451 | 453 | 2 | 0.90 | 0.00 |

(b) Restricted dataset

| Registration Method | Pyramid Level | Use Scale Estimate | Logpolar Res | Valid Truth | Avg Location MAE | Std Location MAE | Success | Valid Est | False Positive | Success Rate | False Positive Rate |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **NGC_OC** | 0,0 | **True** | 512x512 | 51 | 6.44 | 15.37 | 44 | 50 | 6 | 0.86 | 0.12 |
| | **1,2** | | **256x256** | **51** | **7.16** | **16.05** | **44** | **50** | **6** | **0.86** | **0.12** |
| | | | 512x512 | 51 | 6.46 | 15.56 | 44 | 49 | 5 | 0.86 | 0.10 |
| | 0,0 | | 512x256 | 51 | 7.58 | 17.02 | 43 | 50 | 7 | 0.84 | 0.14 |
| | | | 256x256 | 51 | 7.57 | 16.89 | 43 | 50 | 7 | 0.84 | 0.14 |
| | 1,1 | | | 51 | 6.93 | 16.09 | 43 | 49 | 6 | 0.84 | 0.12 |
| | | | 512x512 | 51 | 6.25 | 15.62 | 43 | 48 | 5 | 0.84 | 0.10 |
| | 1,2 | | 512x256 | 51 | 7.19 | 16.32 | 43 | 49 | 6 | 0.84 | 0.12 |
| | 2,2 | | 256x256 | 51 | 7.11 | 15.91 | 43 | 50 | 7 | 0.84 | 0.14 |
| | | | 512x256 | 51 | 7.24 | 16.18 | 43 | 50 | 7 | 0.84 | 0.14 |

(c) Benchmark dataset

| Registration Method | Pyramid Level | Use Scale Estimate | Logpolar Res | Valid Truth | Avg Location MAE | Std Location MAE | Success | Valid Est | False Positive | Success Rate | False Positive Rate |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **NGC_OC** | 1,2 | **True** | 512x512 | 295 | 2.19 | 1.94 | 294 | 294 | 0 | 1.00 | 0.00 |
| | 1,1 | | | 295 | 1.06 | 0.76 | 294 | 294 | 0 | 1.00 | 0.00 |
| | **1,2** | | 512x256 | 295 | 2.21 | 1.94 | 293 | 293 | 0 | 0.99 | 0.00 |
| | | | **256x256** | **295** | **2.19** | **1.94** | **293** | **293** | **0** | **0.99** | **0.00** |
| | 1,1 | | 512x256 | 295 | 1.09 | 0.75 | 293 | 293 | 0 | 0.99 | 0.00 |
| | | | 256x256 | 295 | 1.08 | 0.76 | 293 | 293 | 0 | 0.99 | 0.00 |
| | 0,0 | | 512x512 | 295 | 0.90 | 0.69 | 293 | 293 | 0 | 0.99 | 0.00 |
| | | | 256x256 | 295 | 0.94 | 0.70 | 292 | 292 | 0 | 0.99 | 0.00 |
| | 2,2 | | | 295 | 2.20 | 1.94 | 290 | 290 | 0 | 0.98 | 0.00 |
| | | | 512x512 | 295 | 2.19 | 1.93 | 290 | 290 | 0 | 0.98 | 0.00 |

Table 4.2: Coarse registration methods with top 10 success rates on each dataset. Bold denotes the coarse registration method and corresponding parameters chosen for use in the hybrid method. The values for valid truth, success, valid est, and false positive correspond to the number of frames with valid truth, the number of frames that are correctly registered, the number of frames that are registered (not necessarily correctly), and the number of frames that are incorrectly registered, respectively.

fraction of the computation time.

| Transform | Images | Proposed NGC_OC $(s,\theta)$ | Proposed NGC_OC $(\hat{S},\hat{\theta})$ | Proposed NGC_OC (no scale) $(s,\theta)$ | Proposed NGC_OC (no scale) $(\hat{S},\hat{\theta})$ | NGC Tzimiropoulos [45] $(s,\theta)$ | NGC Tzimiropoulos [45] $(\hat{S},\hat{\theta})$ |
|---|---|---|---|---|---|---|---|
| Rotation | Mars | (1.01,39.45) | (1.01,39.29) | (1.01,39.45) | (1.01,39.30) | - | - |
| | Monet | (1.01,39.45) | (1.01,39.20) | (1.01,39.45) | (1.01,39.20) | - | - |
| | New York | (1.00,169.91) | (1.00,169.91) | (1.00,169.91) | (1.00,169.91) | - | - |
| | Van Gogh | (1.00,167.21) | (1.00,167.23) | (1.00,167.21) | (1.00,167.23) | - | - |
| Rot. & Scale | Bark | (4.00,210.16) | **(4.00,210.02)** | (4.00,210.16) | **(4.00,210.00)** | (4.09,153.4) | **(4.01,150.1)** |
| | Boat | (4.27,47.12) | **(4.28,45.68)** | (4.27,47.12) | **(4.28,45.64)** | (4.36,46.0) | **(4.26,45.7)** |
| | East Park | (5.76,0.17) | **(5.76,359.92)** | (5.76,0.17) | **(5.76,359.77)** | (5.77,0.6) | **(5.78,0.4)** |
| | East South | (5.19,299.69) | **(5.15,300.54)** | (5.19,299.69) | **(5.15,300.47)** | (5.09,60.0) | **(5.18,59.4)** |
| | Ensimag | (5.82,342.05) | **(5.81,329.75)** | (4.66,42.94) | (4.76,41.64) | (4.92,40.7) | (4.76,41.5) |
| | Inria | (5.80,1.01) | **(5.78,0.34)** | (3.91,358.35) | (3.91,0.73) | (4.03,0.8) | (3.91,0.7) |
| | Inria Model | (5.57,340.94) | **(5.55,340.33)** | (4.04,25.21) | (4.05,24.85) | (4.79,50.82) | (4.82,51.0) |
| | Laptop | (1.51,315.32) | **(1.51,314.86)** | (1.51,315.32) | **(1.51,314.83)** | (1.51,45.4) | **(1.51,45.0)** |
| | Resid | (5.88,324.71) | **(5.85,328.42)** | (5.88,324.71) | **(5.84,328.41)** | (5.89,33.2) | **(5.85,31.6)** |
| | Ubc | (2.86,350.39) | **(2.87,350.36)** | (2.86,350.39) | **(2.88,350.43)** | (2.89,9.6) | **(2.89,9.5)** |
| Scale | Asterix | (5.79,0.06) | **(5.77,0.11)** | (4.51,359.90) | (4.49,0.00) | (6.0,0.0) | **(5.78,0.0)** |
| | Belledonne | (5.61,1.63) | **(5.60,0.08)** | (5.61,1.63) | **(5.60,359.90)** | (5.34,0.0) | **(5.57,0.35)** |
| | Bip | (3.74,359.13) | **(3.73,359.93)** | (3.73,0.11) | **(3.73,359.99)** | (3.75,0.0) | (3.73,0.0) |
| | Crolle | (4.67,0.20) | **(4.73,0.36)** | (3.97,0.11) | (4.02,0.52) | (4.01,0.0) | (3.97,0.7) |
| | Laptop | (6.24,359.45) | **(6.22,359.73)** | (6.24,359.45) | **(6.22,359.74)** | (6.25,0.0) | **(6.22,0.35)** |
| | Van Gogh | (5.71,0.16) | **(5.75,359.97)** | (2.80,0.16) | (2.81,359.93) | (3.4,0.0) | (3.38,0.0) |
| Blur | Bikes | (1.03,0.64) | (1.03,0.41) | (1.02,0.54) | (1.02,0.43) | - | - |
| | Trees | (1.02,2.95) | (1.02,3.20) | (1.02,2.57) | (1.02,2.78) | - | - |
| Compression | Ubc | (1.00,0.00) | (1.00,360.00) | (1.00,0.00) | (1.00,0.00) | - | - |
| Illumination | Cars | (1.00,359.91) | (1.00,0.29) | (1.00,359.91) | (1.00,359.99) | - | - |

Table 4.3: Maximum scale recovered by each registration method on benchmark dataset. The maximum scale recovered across all three methods is shown in bold. Note that the convention used for rotation is positive clockwise, but the baseline rotations are provided as positive counter-clockwise values.

### 4.3.2 Proposed Hybrid Method Comparison to KLT and Feature-based Methods

The second experiment compares the performance of the proposed hybrid method and its variants to KLT and several feature-based methods. Results on the public, restricted, and benchmark datasets are shown in Figs. 4.13 to 4.15, Figs. 4.16 to 4.18, and Figs. 4.19 to 4.21, respectively. Robustness is measured by success rate and false positive rate metrics. Accuracy is measured by mean absolute error in pixel location. For robustness plots, success rate and false positive rate, the x-axis contains the location MAE threshold of the

corresponding y-axis value. In other words, the y-axis robustness metric is computed for all cases where the location MAE metric is less than the given threshold on the x-axis.

In no particular order, the feature-based methods selected for comparison are SIFT, SURF, FREAK, LATCH, and A-KAZE. In order to achieve acceptable performance on the public and restricted datasets, the sensitivity of the detector for each feature descriptor had to be increased in order to detect over 6,000 features in each image. This has the undesirable consequence of significantly increasing computation time to the point that even the real-time capable descriptors of FREAK, LATCH, and A-KAZE are no longer capable of real-time performance on these datasets. This behavior is most likely a result of the data, which is highly compressed and a large number of image pairs are devoid of salient features.

KLT is typically the most selective method, having the lowest success rate and the lowest false positive rate, with the exception of the restricted dataset where KLT exhibited one of the highest false positive rates. The results also show that KLT has the lowest average location MAE on the public and benchmark datasets, but falls in the middle of the pack on the restricted dataset.

The proposed scale space search enabled NGC_NCC consistently has the highest success rate on all datasets. NGC_NCC has a lower false positive rate than all other methods on the restricted dataset, a false positive rate that is comparable to the best solution on the benchmark dataset, and is slightly outperformed by KLT, SIFT, and SURF false positive rates on the public dataset. NGC_NCC has slightly higher average location MAE than other competing methods, such as KLT.

The increase in success rate and decrease in false alarm rate as a result of using scale space search is most apparent in the results for the restricted and benchmark datasets, which contain a large number of high scale factor test cases. Overall, the results show that NGC_NCC significantly outperforms competing methods in terms of success rate, has a reasonable, but data dependent false positive rate, and is on the order of approximately one

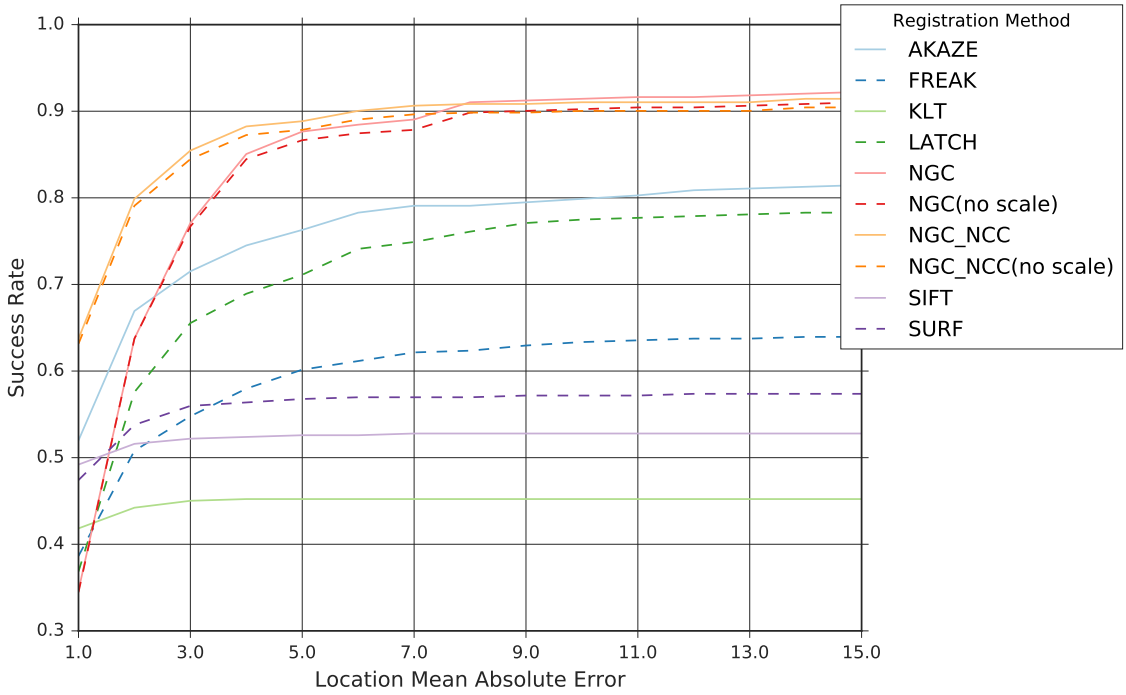



Figure 4.13: Success rate of proposed hybrid method, KLT, and feature-based methods on public dataset

to two pixels in location MAE less accurate than KLT.

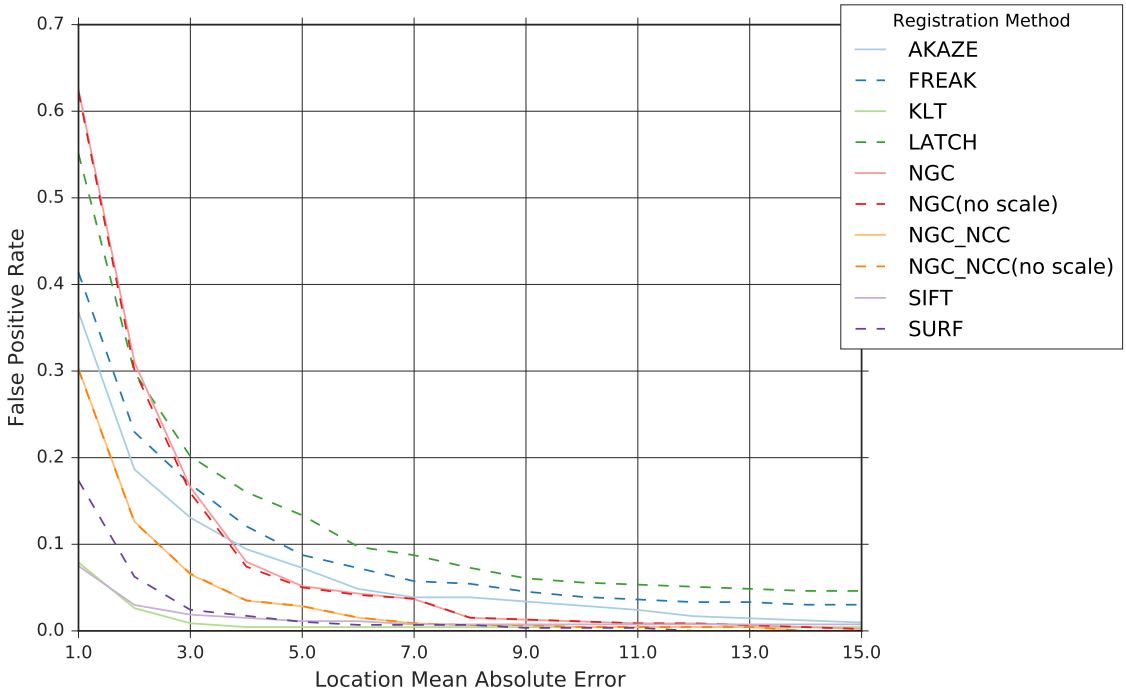



Figure 4.14: False positive rate of proposed hybrid method, KLT, and feature-based methods on public dataset

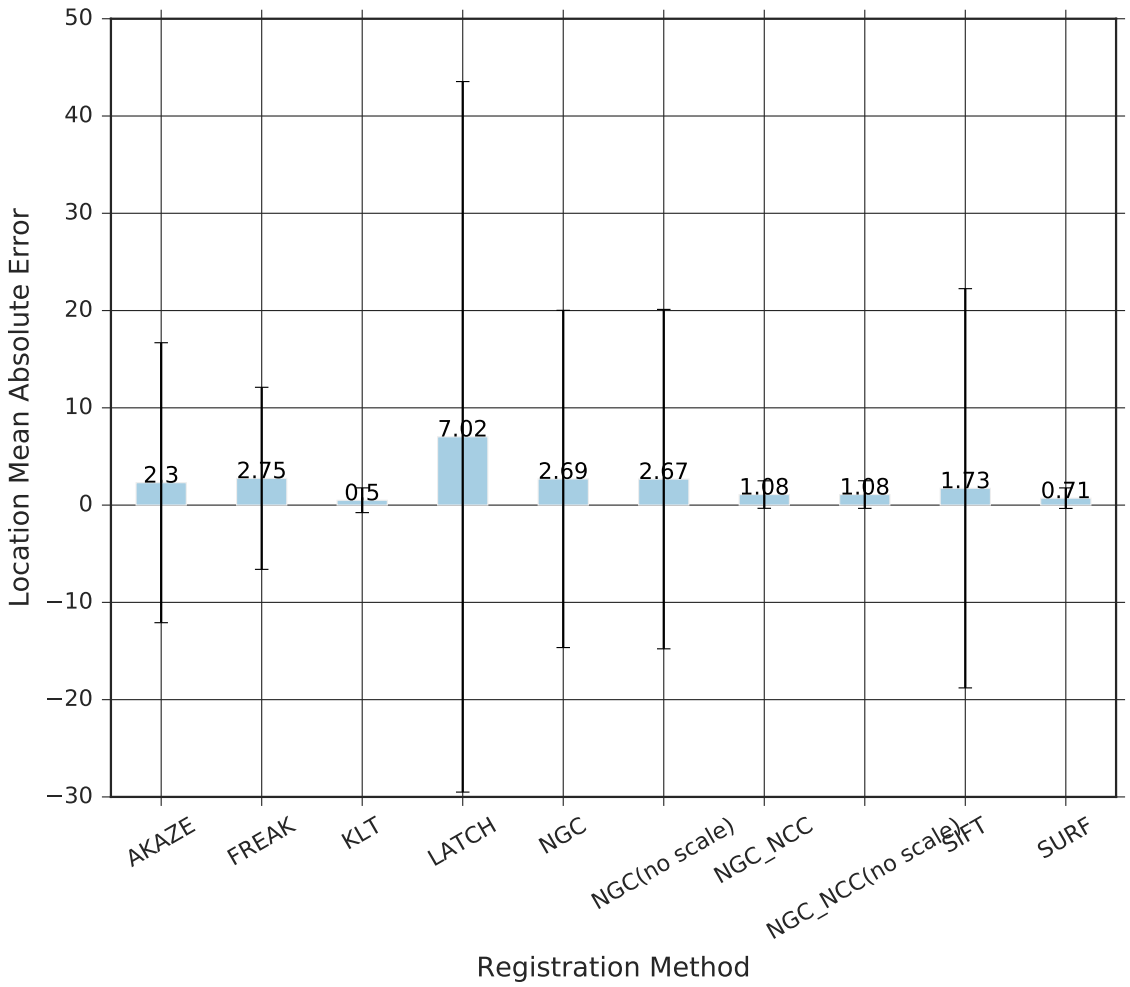



Figure 4.15: Accuracy of proposed hybrid method, KLT, and feature-based methods on public dataset

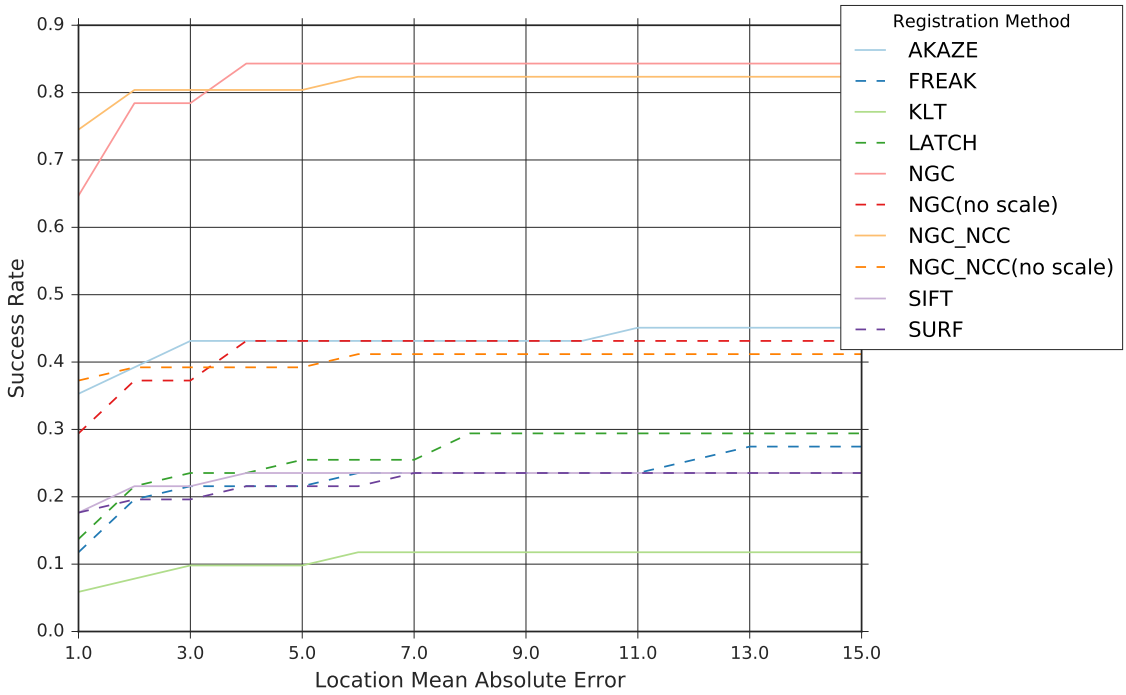



Figure 4.16: Success rate of proposed hybrid method, KLT, and feature-based methods on restricted dataset

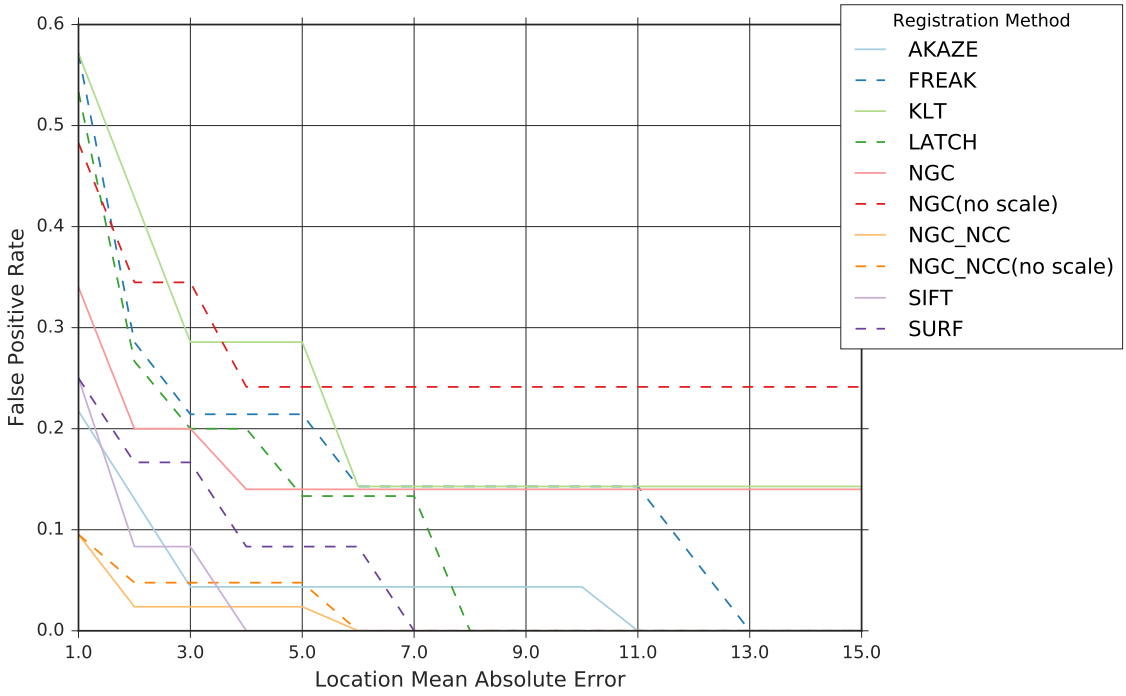



Figure 4.17: False positive rate of proposed hybrid method, KLT, and feature-based methods on restricted dataset

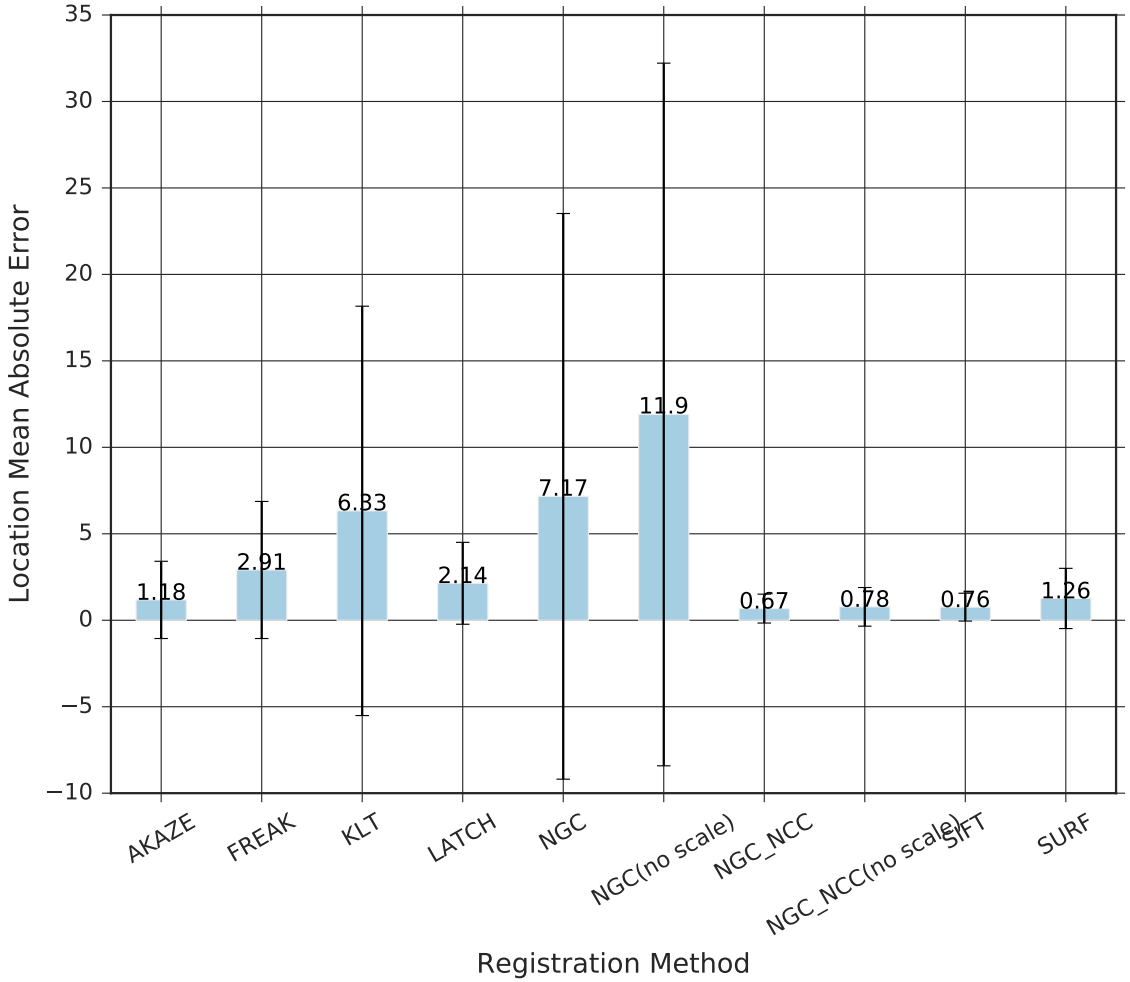



Figure 4.18: Accuracy of proposed hybrid method, KLT, and feature-based methods on restricted dataset

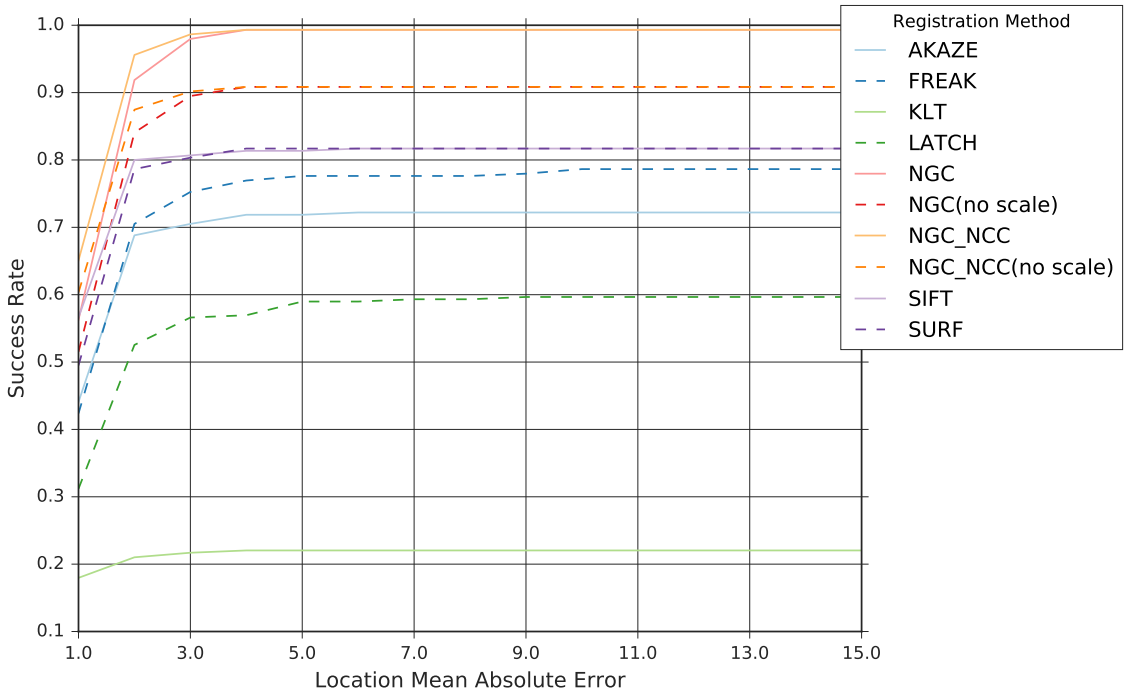



Figure 4.19: Success rate of proposed hybrid method, KLT, and feature-based methods on benchmark dataset

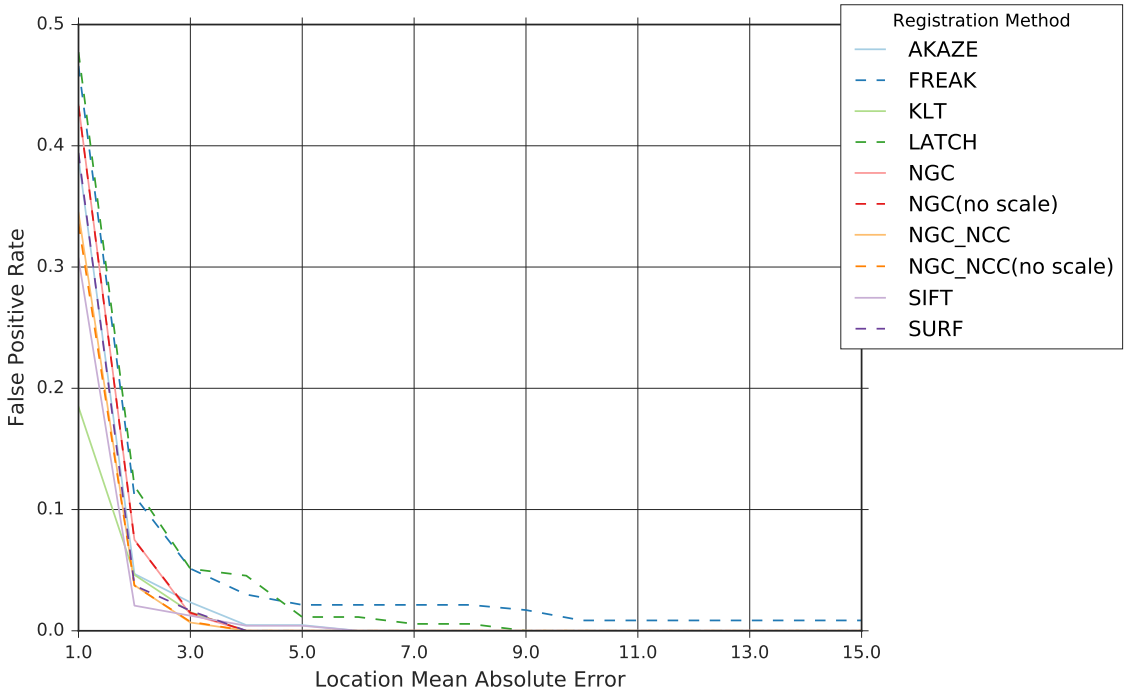



Figure 4.20: False positive rate of proposed hybrid method, KLT, and feature-based methods on benchmark dataset

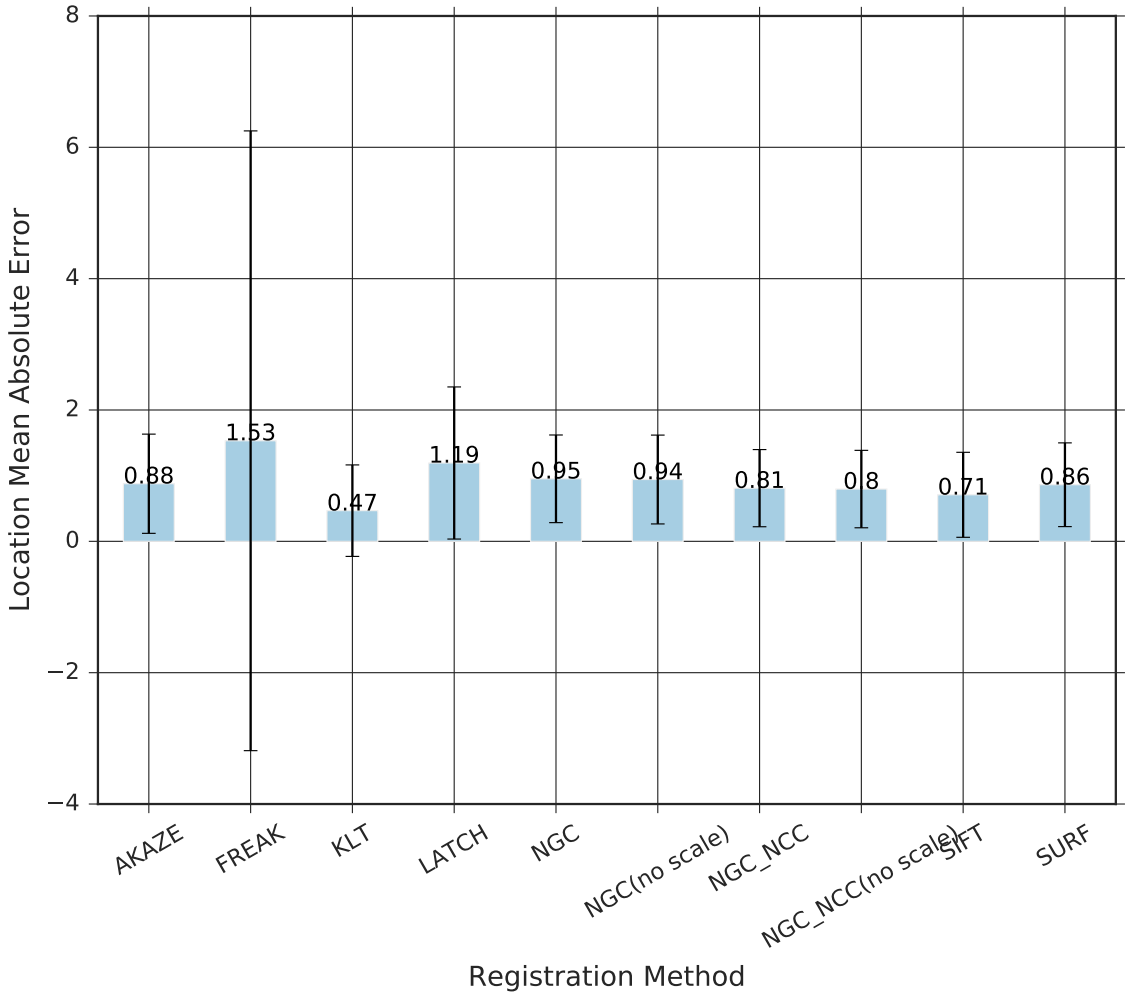



Figure 4.21: Accuracy of proposed hybrid method, KLT, and feature-based methods on benchmark dataset

### 4.3.3 Proposed Hybrid Method Comparison to KLT and Optimization Methods

The third experiment is the same as the second experiment, but instead compares the performance of the proposed hybrid method and its variants to hybrid methods using optimization as an alternative to grid-based NCC for fine registration. Robustness and accuracy results on the public, restricted, and benchmark datasets are shown in Figs. 4.22 to 4.24, Figs. 4.25 to 4.27, and Figs. 4.28 to 4.30, respectively. For plots of robustness metrics, success rate and false positive rate, the x-axis contains the location MAE threshold of the corresponding y-axis value. In other words, the y-axis robustness metric is computed for all cases where the measured location MAE is less than the given threshold on the x-axis.

Four hybrid optimization methods were evaluated that combine NGC and either efficient second order minimization (ESM) or inverse compositional Lucas-Kanade (ICLK) optimization with normalized cross correlation (NCC) or sum of squared difference (SSD) objective functions. The optimization implementations were obtained from the modular tracking framework (MTF) [67] developed by Singh and Jagersand. The MTF implementation was modified to incorporate binary image masks to enable tracking (registering) the entire image while ignoring non-overlapping regions resulting from coarse registration. Each evaluated hybrid NGC-optimization utilizes the proposed scale space search used by NGC_NCC.

The NGC_ESM_NCC algorithm performed the best of the four hybrid optimization variants in terms of success rate, false positive rate, and average location MAE. The success rate and accuracy of NGC_ESM_NCC was comparable to NGC_NCC only on the benchmark dataset, but slightly worse than NGC_NCC on the other two datasets. One possible explanation for why NGC_ESM_NCC performed slightly better on the benchmark dataset again comes down to image noise. It is likely that the optimization converges reliably and accurately on pristine benchmark images, but has a slightly lower converge rate and

accuracy on the noisy and compressed public and restricted video data. The higher than expected false positive rates and location MAE of NGC_ESM_NCC on all three datasets can be attributed to the fact that no method for failure determination was implemented for any of the optimization-based hybrid methods. This check could be easily added in the future by checking to see if the optimization method converged to within a predetermined threshold based on the objective function. Overall, NGC_NCC still outperforms the best optimization-based hybrid method by approximately 10 percent in success rate. The best optimization-based hybrid methods are still viable alternatives to grid-based NCC, but additional experimentation is needed to determine when and why optimization fails where grid-based NCC succeeds.

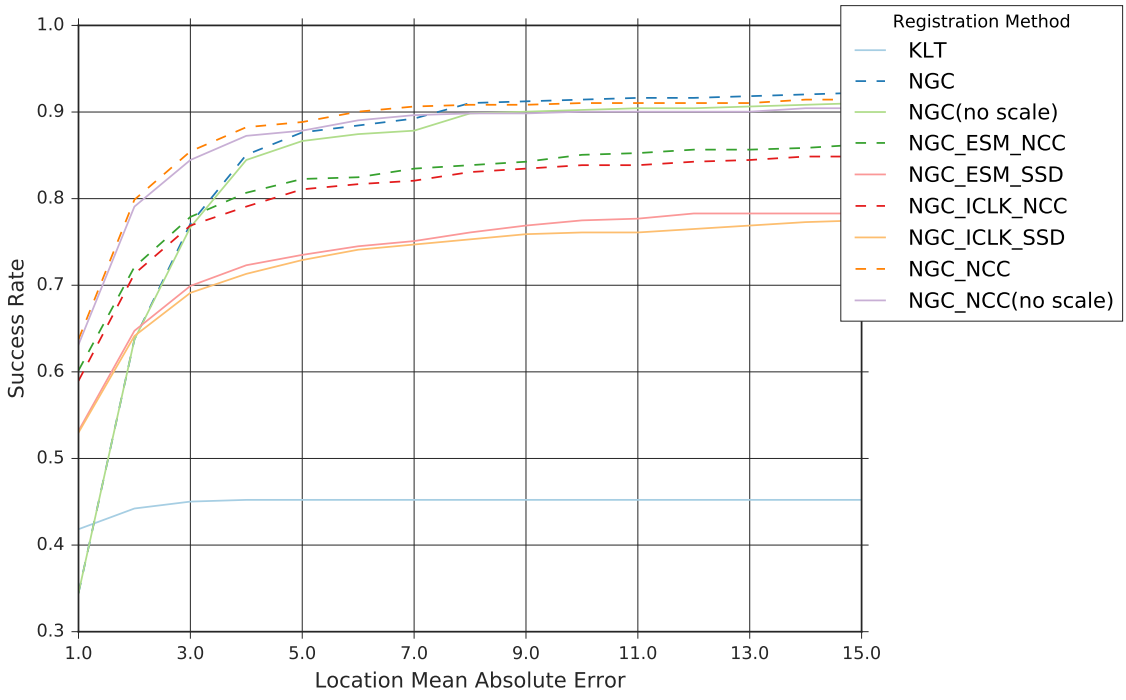



Figure 4.22: Success rate of proposed hybrid method, KLT, and optimization methods on public dataset.

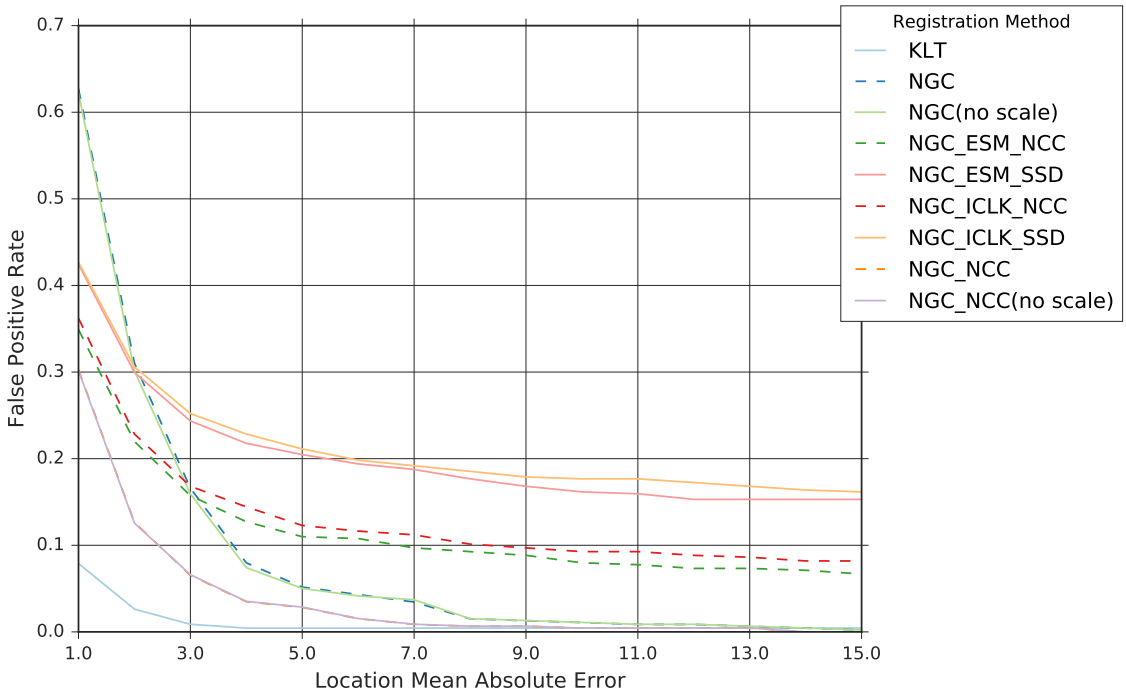



Figure 4.23: False positive rate of proposed hybrid method, KLT, and optimization methods on public dataset.

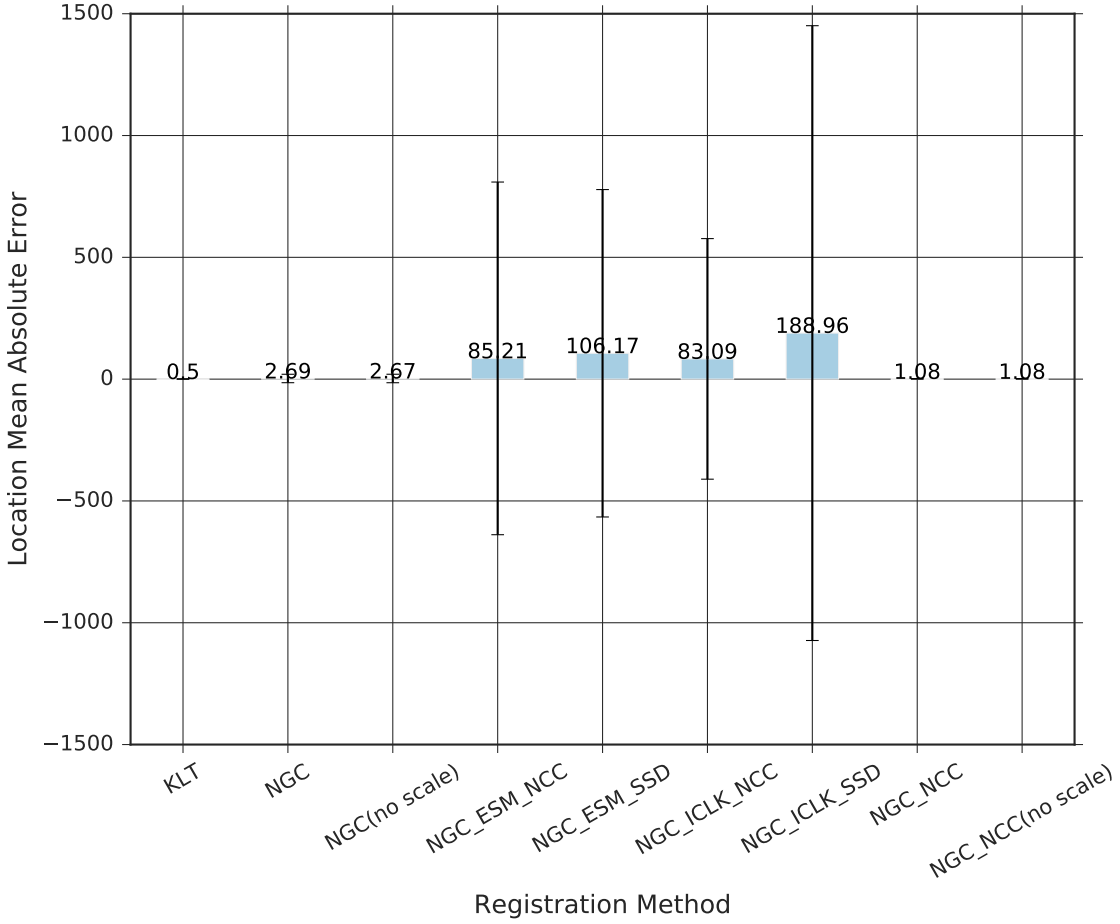



Figure 4.24: Accuracy of proposed hybrid method, KLT, and optimization methods on public dataset.

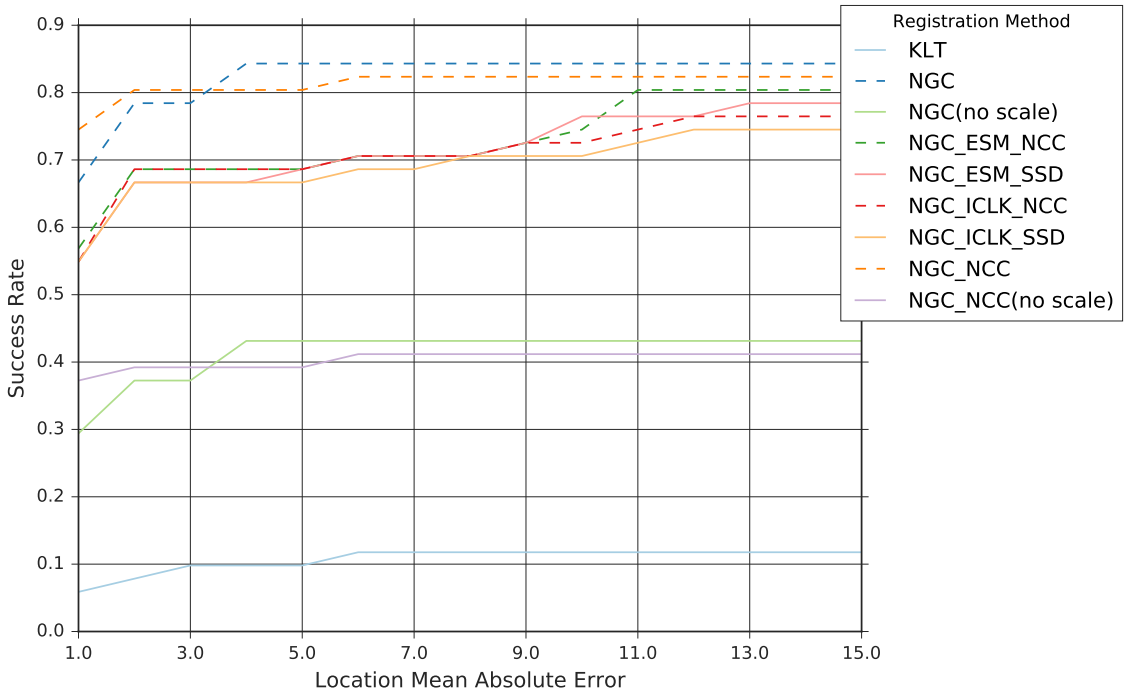



Figure 4.25: Success rate of proposed hybrid method, KLT, and optimization methods on restricted dataset.

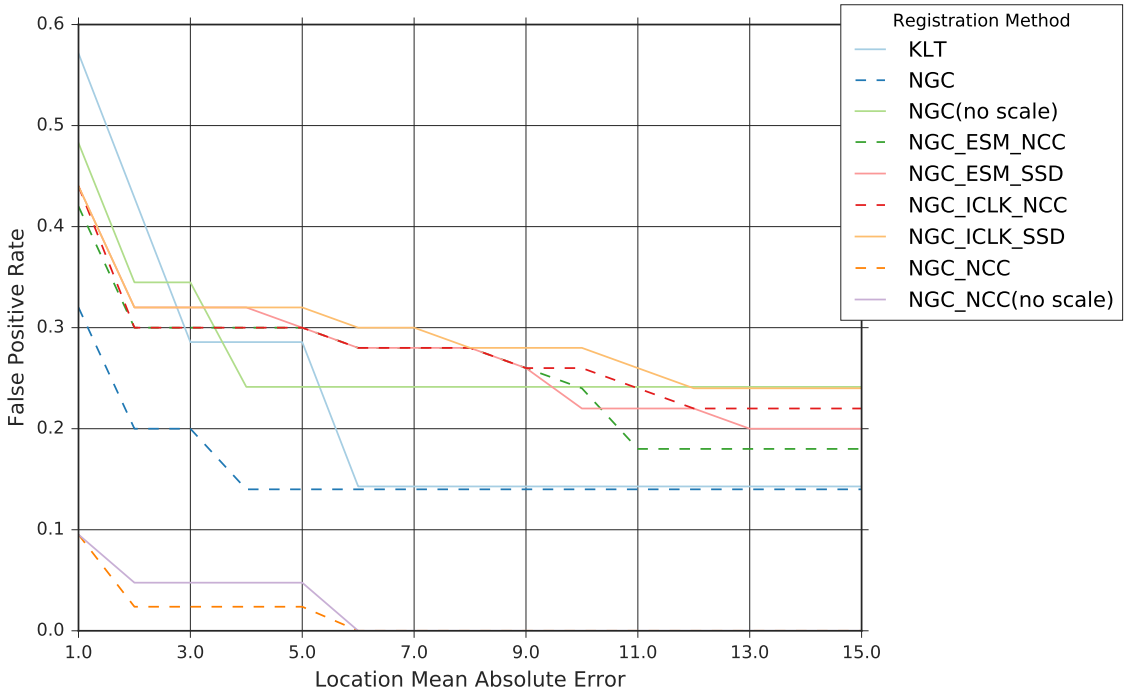



Figure 4.26: False positive rate of proposed hybrid method, KLT, and optimization methods on restricted dataset.

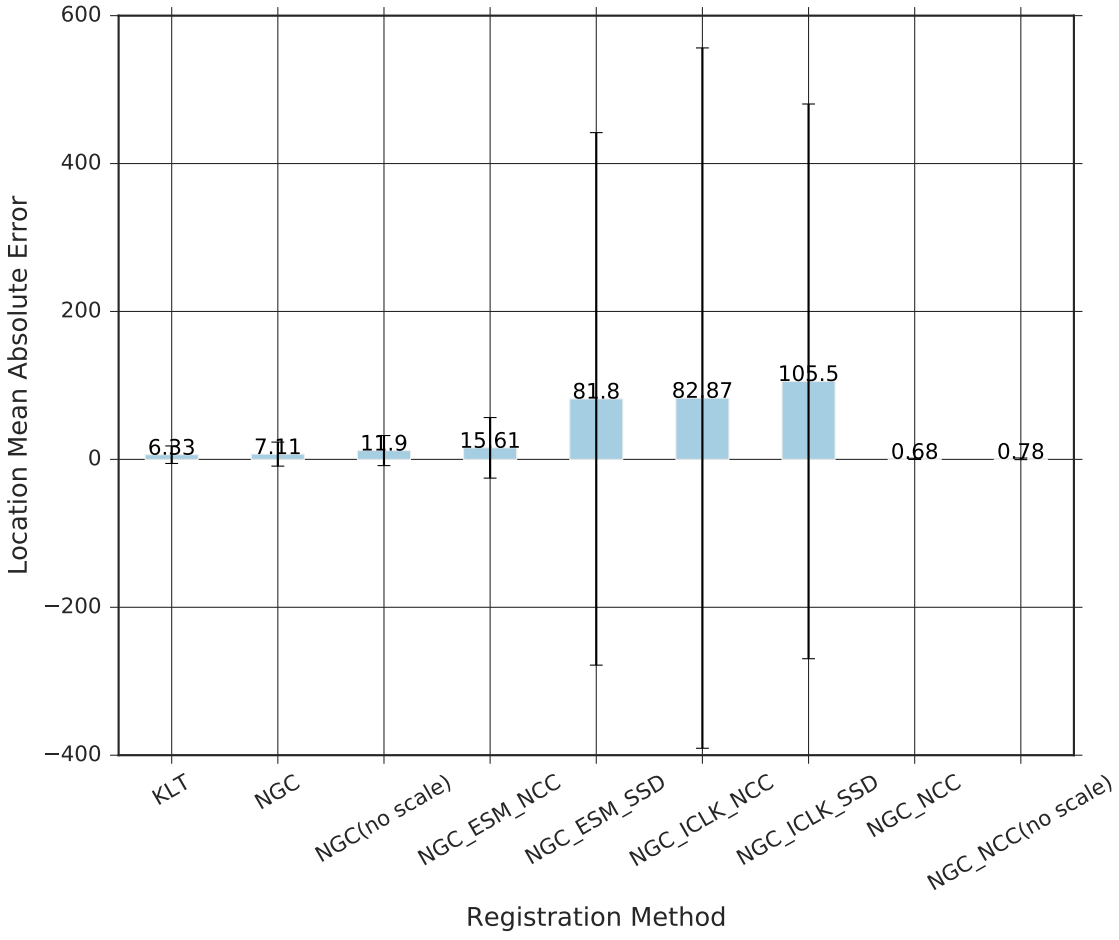



Figure 4.27: Accuracy of proposed hybrid method, KLT, and optimization methods on restricted dataset.

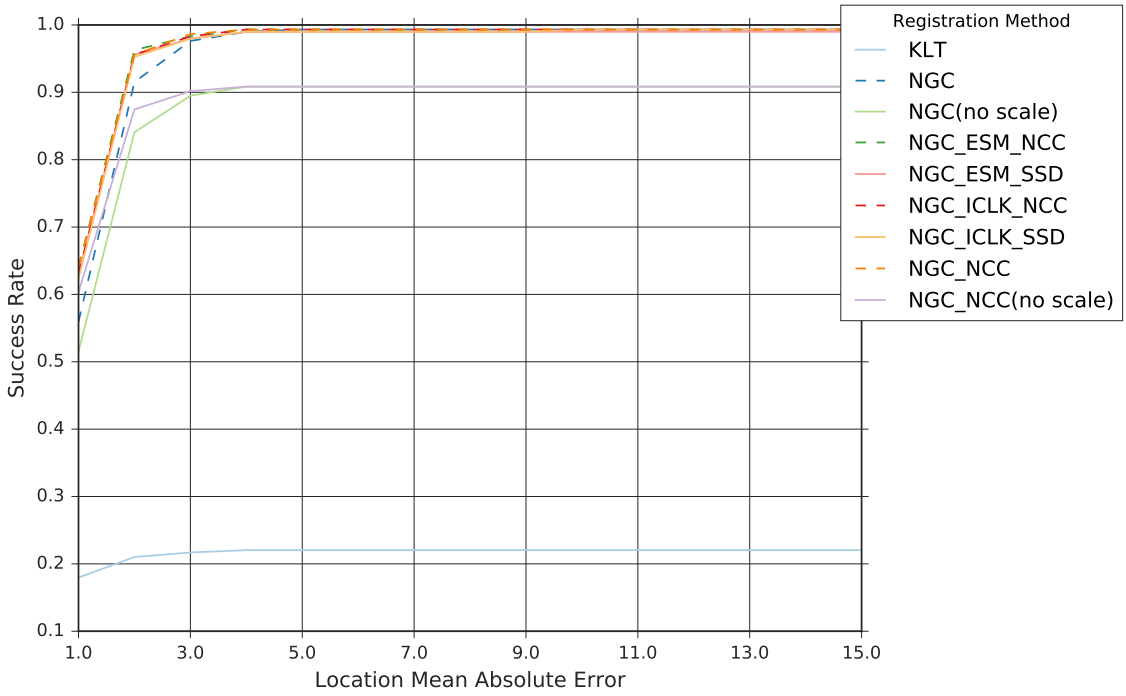



Figure 4.28: Success rate of proposed hybrid method, KLT, and optimization methods on benchmark dataset.

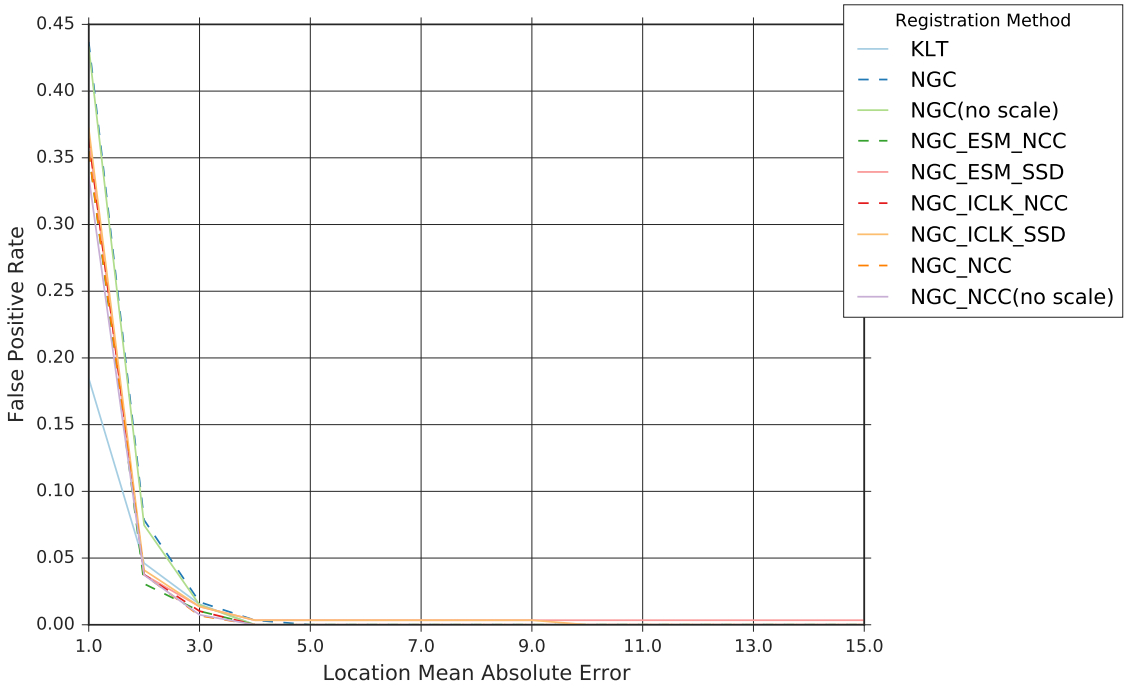



Figure 4.29: False positive rate of proposed hybrid method, KLT, and optimization methods on benchmark dataset.

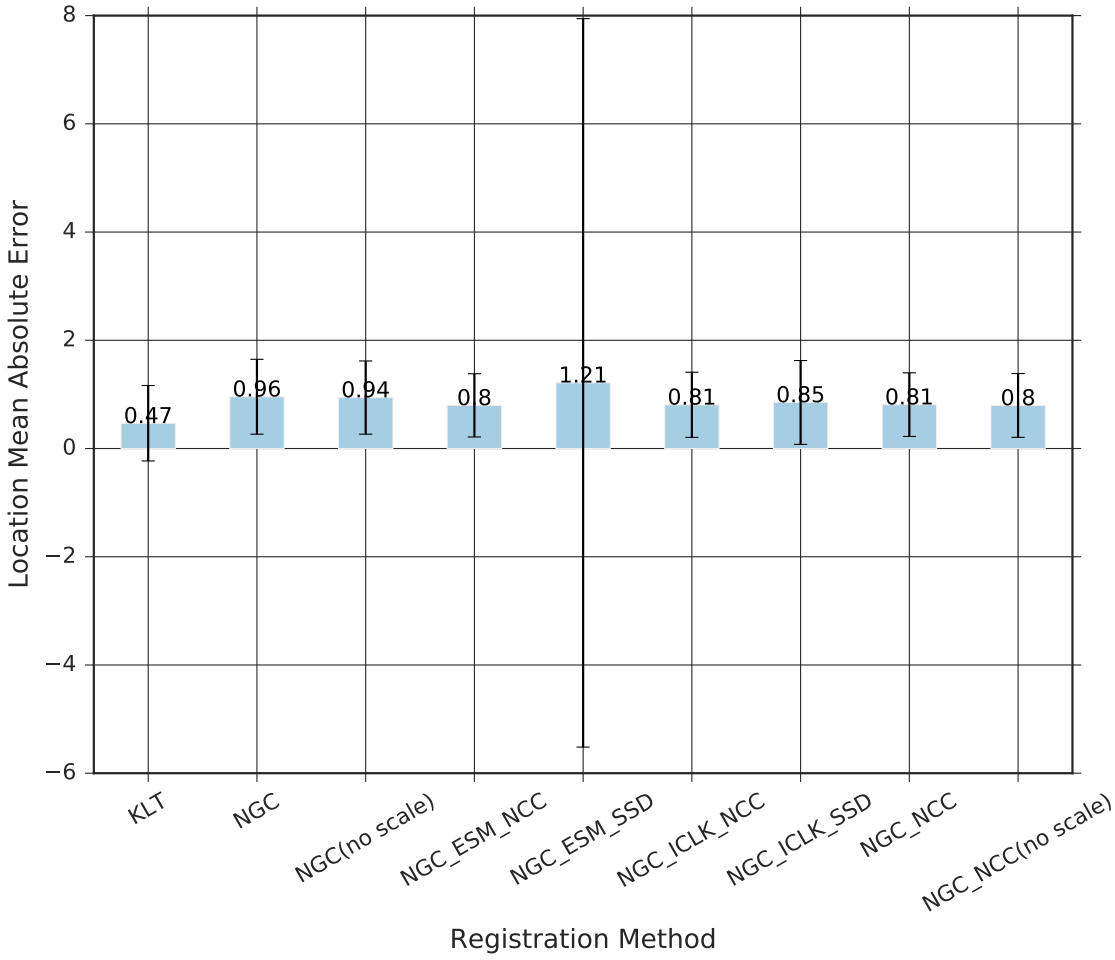



Figure 4.30: Accuracy of proposed hybrid method, KLT, and optimization methods on benchmark dataset.

### 4.3.4 Accuracy and Robustness Evaluation of Proposed Method on Long Video Sequences

The accuracy and robustness of the proposed method and KLT were evaluated on the longer contiguous video sequences described in Section 4.1. As previously mentioned, these datasets do not have truth homography data; so robustness was evaluated based on algorithm reported success/failure status as discussed in Section 4.2 and accuracy was evaluated using Eq. (4.2). Keeping this in mind, the results in Table 4.4 show that when compared to KLT, the proposed hybrid NGC_NCC method, both with and without scale space search, registers significantly longer average and maximum contiguous video sequences with fewer shot breaks (contiguous failures over one or more frames) on all test video sequences. It can also be observed that the scale space enhanced NGC_NCC method yields a small improvement to robustness over the same method without scale space search. This small difference in this robustness can be attributed to the small percentage of video frames that contain large scale differences. These results also indicate that KLT is more accurate than NGC_NCC as KLT yields lower average intensity MAE than NGC_NCC on all test sequences. These results are consistent with earlier results and conclusions that NGC_NCC is significantly more robust than KLT, but this robustness comes at the cost of reduced accuracy.

| Dataset | Registration Method | Avg Intensity MAE | Std Intensity MAE | Total Frames | Frames Registered | Max Contiguous Frames Registered | Avg Contiguous Frames Registered | Shot Breaks |
|---|---|---|---|---|---|---|---|---|
| EO Video Sequence 1 | KLT | 1.03 | 0.31 | 59883 | 59482 | 21744 | 1044 | 56 |
| | NGC | 1.17 | 0.42 | 59883 | 59857 | 45933 | 14964 | 3 |
| | NGC(no scale) | 1.17 | 0.42 | 59883 | 59856 | 45933 | 11971 | 4 |
| | NGC_NCC | **1.15** | 0.36 | 59883 | 59830 | **43587** | **3519** | **16** |
| | NGC_NCC(no scale) | **1.15** | 0.36 | 59883 | 59829 | **43587** | **3519** | **16** |
| EO Video Sequence 2 | KLT | 1.55 | 0.84 | 42580 | 41538 | 5546 | 103 | 401 |
| | NGC | 1.97 | 1.01 | 42580 | 42483 | 19553 | 924 | 45 |
| | NGC(no scale) | 1.97 | 1.01 | 42580 | 42471 | 11635 | 849 | 49 |
| | NGC_NCC | **1.80** | 0.95 | 42580 | 42250 | **11077** | **464** | **90** |
| | NGC_NCC(no scale) | **1.80** | 0.95 | 42580 | 42242 | **8398** | **449** | **93** |
| IR Video Sequence | KLT | 1.66 | 0.48 | 20000 | 19315 | 3152 | 276 | 69 |
| | NGC | 2.01 | 0.96 | 20000 | 19832 | 5887 | 381 | 51 |
| | NGC(no scale) | 2.01 | 0.94 | 20000 | 19819 | 4060 | 342 | 57 |
| | NGC_NCC | **1.75** | 0.72 | 20000 | 19715 | **4057** | **379** | **51** |
| | NGC_NCC(no scale) | **1.74** | 0.72 | 20000 | 19706 | **4057** | **352** | **55** |
| Restricted Video Sequence | KLT | 3.86 | 0.95 | 4195 | 4146 | 1578 | 259 | 15 |
| | NGC | 4.08 | 1.88 | 4195 | 4173 | 1586 | 596 | 6 |
| | NGC(no scale) | 4.08 | 1.88 | 4195 | 4173 | 1586 | 596 | 6 |
| | NGC_NCC | **4.02** | 1.49 | 4195 | 4169 | **1584** | **521** | **7** |
| | NGC_NCC(no scale) | **4.02** | 1.49 | 4195 | 4169 | **1584** | **521** | **7** |

Table 4.4: Accuracy and robustness of proposed coarse and hybrid methods compared to KLT on long video sequences. Bold values are used to emphasize to the NGC_NCC method with and without scale space search and underlined values to emphasize KLT.

### 4.3.5 Average Execution Time

The average execution time for each method was measured on EO Video Sequence 1 of the Public Long Dataset, which is representative of typical aerial video sequences. Execution times were computed on an Intel i7 3940 XM CPU, which is a 3rd generation Ivy Bridge mobile processor. As shown in Table 4.5, the proposed coarse registration method, NGC, requires 26.7 ms with scale space search or 9.6 ms without scale space search. The proposed hybrid registration method, NGC_NCC, requires 45.9 ms with scale space search. This corresponds to approximately 22 Hz performance, which falls short of real-time for 30 Hz video. The proposed hybrid method, NGC_NCC (no scale), performs faster than real-time at 27.5 ms without scale space search. In comparison, the KLT implementation, which is built upon highly optimized methods from OpenCV, is also faster than real-time with an average execution time of 23.3 ms.

| | NGC | NGC (no scale) | NGC_NCC | NGC_NCC (no scale) | KLT |
|---|---|---|---|---|---|
| execution time (ms) | 26.7 | 9.6 | 45.9 | 27.5 | 23.3 |

Table 4.5: Average execution time of proposed method compared to KLT. Execution time was measured over 1000 iterations and excludes time spent reading images from disk.

There are still opportunities to optimize the implementation of NGC_NCC and its variants to further improve performance. One significant optimization would be to avoid duplicate computation of pixel products within overlapping template regions of grid-based normalized cross correlation. Another optimization would be to reuse the image pyramid, complex gradient image, and spectrum magnitude of the FMT of the complex gradient image from the destination frame as the next source frame when processing consecutive frames in video.

# Summary

## 5.1 Concluding Remarks

**The contributions of this thesis focus on providing a complementary or alternative method to KLT for aerial video image registration in the presence of large displacement and other challenging conditions.** A hybrid coarse-fine video image registration method was developed to improve robustness to large displacement transformations. The speed, accuracy, and robustness of the proposed hybrid method was evaluated and compared to KLT, optimization methods, and feature-based methods. A multi-resolution scale space search was developed and incorporated into the proposed hybrid method to enable processing reduced resolution images with improved speed and robustness, particularly with respect to large scale factors. The proposed hybrid registration method was implemented in C++, multi-threaded, and optimized to achieve real-time processing on video sequences containing scale factors up to 2. An adaptive peak-to-sidelobe ratio threshold test was employed to improve the ability of the proposed method to detect registration failure.

Results demonstrate the viability of the proposed hybrid method for aerial video image registration. Mores specifically, results show that the proposed method is significantly more robust than KLT, optimization methods, and feature-based methods on three challenging evaluation datasets. This improved robustness is achieved in exchange for a small reduction in accuracy compared to KLT. Execution time was evaluated and the proposed hybrid method was able to achieve over 30 Hz for scale factors up to 2 and over 20 Hz for scale factors up

to 6. Finally, results show that the scale space search enhanced coarse registration method is able to more consistently recover large scale factors up to and sometimes exceeding 6 while requiring less computation time than the baseline coarse method.

In addition to the primary contribution of a new hybrid algorithm, several lessons were learned over the coarse of this thesis. Three limiting factors to Fourier-Mellin based image registration using the fast Fourier transform were realized. First, non-overlapping image content can significantly diminish signal strength in normalized gradient correlation, sometimes below the noise floor. This effect is more pronounced in the presence of larger scale factors. Second, aliasing due to the finite fast Fourier transform and image sampling at different scales for large scale factors can negatively impact registration performance. Performing correlation on gradient images as well as searching over multiple scales helps alleviate aliasing, but the problem still exists. Third, the non-uniformity in log polar sampling can also affect registration performance. Multiple alternative methods and sampling strategies were explored, but either their anecdotal performance or computation time could not be justified.

Achieving comparable accuracy to the Kanade-Lucas-Tomasi (KLT) feature detection and tracking algorithm is a difficult task given the challenging conditions present in the tested public and restricted datasets. Further testing and analysis is necessary to determine if the accuracy discrepancy between the proposed hybrid registration method and KLT is a limitation of the proposed method or caused by specific conditions present in the data. This issue is further complicated by the fact that accuracy results are averaged over all valid registration attempts and the proposed method succeeds on a significantly larger number of test cases than KLT. Investigating the causes of the accuracy discrepancy as well as exploring methods to improve accuracy are left for future work.

## 5.2 Future Work

There are several avenues that could be explored in the future in order to better characterize behavior of the investigated registration methods and to improve speed, accuracy, and robustness of the proposed hybrid registration method.

The alternative scale space search method presented in Fig. 3.7 could be quantitatively compared to the proposed scale space search in Fig. 3.8 instead of the qualitative comparison conducted. A quantitative evaluation would provide a more thorough understanding and comparison of the two proposed scale space search methods.

There are several implementation optimizations that could be made to further improve the speed of NGC_NCC and possibly achieve real-time performance when scale space search is enabled. First, when registering consecutive frames in video, the destination image pyramid, gradient images, FMT spectrum magnitude, and gradient of the log polar transformed spectrum magnitude can all be saved and reused as the source for the next consecutive frame to be processed. Second, due to the overlap of template and search regions in grid-based NCC, there are a significant number of redundant computations performed. These computed values could be stored in memory to avoid repeating the calculation in order to reduce computation time.

In an effort to improve accuracy, additional investigation of hybrid registration methods combining NGC and optimization methods is necessary. In theory optimization methods are capable of comparable or better accuracy than KLT feature detection and tracking, which performs a Newton-Raphson type optimization of two parameters to estimate pure translation for each patch. First, it is important to understand why optimization sometimes fails to converge when used as a refinement step in the proposed alternate hybrid NGC methods. Failure is likely caused by the optimization getting trapped in a local minimum or diverging due to content in the image, such as noise, compression artifacts, or low saliency regions. Second, possible solutions must be explored. One possible improvement to convergence might be to perform optimization using patches as in KLT, but instead

select patches from a uniform grid instead of first detecting corner points. Another possible improvement might be to compute an inexpensive measure of saliency at each pixel and suppress or remove pixels with insufficient saliency from the optimization computation. The last idea to explore is hierarchical coarse-to-fine optimization. Even though a coarse estimate is already provided by the proposed coarse registration method, performing optimization first at lower resolutions may improve convergence. If a solution to the convergence issues on the tested datasets is found, it is possible that optimization may yield better performance than grid-based NCC when used as a refinement step in a hybrid registration method.

In order to improve robustness, multiple levels of the scale space pyramid images could be used for coarse OC-based translation estimation to reduce possible aliasing issues caused by the upsampling the lower resolution image to compensate for estimated rotation and scale.

Deep learning has recently demonstrated human level performance on some detection and recognition problems. At least one method for applying deep learning to image homography estimation has already been proposed. Deep learning could be investigated as a potential approach to aerial video image registration. There is limited knowledge available on the limitations of deep learning when applied to image registration so it would be important to conduct a study on deep learning based methods to better understand these limitations and how to improve performance on aerial video image registration.

# References


[1] C. Tomasi and T. Kanade, "Detection and tracking of point features technical report cmu-cs-91-132," *Image Rochester NY*, vol. 91, pp. 1–22, 1991. DOI: 10.1016/S0031-3203(03)00234-6.

[2] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, vol. 81, 1981, pp. 674–679.

[3] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[4] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003. DOI: 10.1016/S0262-8856(03)00137-9.

[5] A. A. Goshtasby, *Image Registration: Principles, Tools and Methods*. London: Springer London, 2012. DOI: 10.1007/978-1-4471-2458-0.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," *Procedings of the Alvey Vision Conference 1988*, pp. 147–151, 1988. DOI: 10.5244/C.2.23.

[7] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, IEEE, 2005, pp. 1508–1515. DOI: 10.1109/ICCV.2005.104.

[8] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, pp. 430–443. DOI: 10.1007/11744023_34.

[9] E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 105–119, 2010. DOI: 10.1109/TPAMI.2008.275.

[10] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *European conference on Computer vision*, 2010, pp. 183–196. DOI: 10.1007/978-3-642-15552-9_14.

[11] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: binary robust invariant scalable keypoints," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2548–2555. DOI: 10.1109/ICCV.2011.6126542.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[13] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, pp. 79 –116, 1998. DOI: 10.1023/A:1008045108935.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–110, 2004. DOI: 10.1023/B:VISI.0000029664.99615.94.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008. DOI: 10.1016/j.cviu.2007.09.014.

[16] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *Lecture Notes in Computer Science (including subseries*

*Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, pp. 778–792. DOI: 10.1007/978-3-642-15561-1_56.

[17] M. Agrawal, K. Konolige, and M. R. Blas, "Censure: center surround extremas for realtime feature detection and matching," in *Computer Vision ECCV 2008*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 102–115. DOI: 10.1007/978-3-540-88693-8_8.

[18] J.-M. Morel and G. Yu, "Asift: a new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, pp. 438–469, 2009. DOI: 10.1137/080732730.

[19] P. F. Alcantarilla, A Bartoli, and A. J. Davison, "Kaze features," *Eur. Conf. on Computer Vision (ECCV)*, pp. 214–227, 2012. DOI: 10.1007/978-3-642-33783-3_16.

[20] J. Dong and S. Soatto, "Domain-size pooling in local descriptors : dsp-sift," no. 1, arXiv: 1412.8556.

[21] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: fast retina keypoint," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 510–517. DOI: 10.1109/CVPR.2012.6247715.

[22] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *British Machine Vision Conference*, 2013. DOI: 10.5244/C.27.13.

[23] G. Levi and T. Hassner, "Latch: learned arrangements of three patch codes," in *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, 2016. DOI: 10.1109/WACV.2016.7477723.

[24] K. Grauman, "The pyramid match kernel : efficient learning with sets of features," *Journal of Machine Learning Research*, pp. 725–760, 2007. DOI: 10.1109/ICCV.2005.239.

[25] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *International Conference on Computer Vision Theory and Applications (VISAPP '09)*, pp. 1–10, 2009. DOI: 10.1.1.160.1721.

[26] M. a. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with," *Communications of the ACM*, vol. 24, pp. 381–395, 1981. DOI: 10.1145/358669.358692.

[27] P. Rousseeuw, "Least median of squares regression," *Journal of the American Statistical Association*, vol. 79, no. 388, pp. 871–880, 1984. DOI: 10.1080/01621459.1984.10477105.

[28] D. Nister, "Preemptive ransac for live structure and motion estimation," in *Proceedings Ninth IEEE International Conference on Computer Vision*, vol. 16, IEEE, 2003, 199–206 vol.1, ISBN: 0-7695-1950-4. DOI: 10.1109/ICCV.2003.1238341.

[29] O. Chum and J. Matas, "Matching with prosac - progressive sample consensus," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 220–226, 2005. DOI: 10.1109/CVPR.2005.221.

[30] R. Raguram, J. M. Frahm, and M. Pollefeys, "A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5303 LNCS, no. PART 2, pp. 500–513, 2008. DOI: 10.1007/978-3-540-88688-4-37.

[31] S. Choi, T. Kim, and W. Yu, "Performance evaluation of ransac family," *Procedings of the British Machine Vision Conference 2009*, pp. 81.1–81.12, 2009. DOI: 10.5244/C.23.81.

[32] F. Hjouj and D. W. Kammler, "Identification of reflected, scaled, translated, and rotated objects from their radon projections," *IEEE Transactions on Image Processing*, pp. 301–310, 2008. DOI: 10.1109/TIP.2007.916160.

[33] Y. Wan and N. Wei, "A fast algorithm for recognizing translated, rotated, reflected, and scaled objects from only their projections," *IEEE Signal Processing Letters*, pp. 71–74, 2010. DOI: 10.1109/LSP.2009.2032487.

[34] N. Nacereddine, S. Tabbone, and D. Ziou, "Similarity transformation parameters recovery based on radon transform. application in image registration and object recognition," *Pattern Recognition*, pp. 2227–2240, 2015. DOI: 10.1016/j.patcog.2015.01.017.

[35] S. Zokai and G. Wolberg, "Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations," *IEEE Transactions on Image Processing*, pp. 1422–1434, 2005. DOI: 10.1109/TIP.2005.854501.

[36] F. Yuan, H. Zhang, and R. Jia, "Digital image stabilization based on log-polar transform," in *Fourth International Conference on Image and Graphics (ICIG 2007)*, IEEE, 2007, pp. 769–773. DOI: 10.1109/ICIG.2007.150.

[37] R. Matungka, Y. Zheng, and R. Ewing, "Image registration using adaptive polar transform," *IEEE Transactions on Image Processing*, pp. 2340–2354, 2009. DOI: 10.1109/TIP.2009.2025010.

[38] R. Matungka, Y. F. Zheng, and R. L. Ewing, "Aerial image registration using projective polar transform," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2009, pp. 1061–1064. DOI: 10.1109/ICASSP.2009.4959770.

[39] M McGuire, "An image registration technique for recovering rotation, scale and translation parameters," *NEC Res. Inst. Tech. Rep., TR*, pp. 1–29, 1998.

[40] B. Srinivasa Reddy and B. N. Chatterji, "An fft-based technique for translation, rotation, and scale-invariant image registration," *IEEE Transactions on Image Processing*, pp. 1266–1271, 1996. DOI: 10.1109/83.506761.

[41] A. Fitch, A. Kadyrov, W. Christmas, and J. Kittler, "Orientation correlation," in *Procedings of the British Machine Vision Conference 2002*, British Machine Vision Association, 2002, pp. 11.1–11.10. DOI: 10.5244/C.16.11.

[42] Y. Keller, A. Averbuch, and M. Israeli, "Pseudo-polar based estimation of large translations rotations and scalings in images," in *Proceedings - IEEE Workshop on Motion and Video Computing, MOTION 2005*, 2007, pp. 201–206, ISBN: 0769522718. DOI: 10.1109/ACVMOT.2005.97.

[43] W. Pan, K. Qin, and Y. Chen, "An adaptable-multilayer fractional fourier transform approach for image registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 400–414, 2009. DOI: 10.1109/TPAMI.2008.83.

[44] Z. Li, J. Yang, M. Li, and R. Lan, "Estimation of large scalings in images based on multilayer pseudopolar fractional fourier transform," 2013.

[45] G. Tzimiropoulos, V. Argyriou, S. Zafeiriou, and T. Stathaki, "Robust fft-based scale-invariant image registration with image gradients," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1899–1906, 2010. DOI: 10.1109/TPAMI.2010.107.

[46] G. Tzimiropoulos and T. Stathaki, "Robust fft-based scale-invariant image registration," *4th SEAS DTC Technical Conferences*, 2010.

[47] G. Tzimiropoulos, V. Argyriou, and T. Stathaki, "Subpixel registration with gradient correlation," *IEEE Transactions on Image Processing*, pp. 1761–1767, 2011. DOI: 10.1109/TIP.2010.2095867.

[48] R Kokila and P Thangavel, "Image registration based on fast fourier transform using gabor filter," *International Journal of Computer Science and Electronics Engineering (IJCSEE)*, vol. 2, no. 1, 2014.

[49] R. Gonzalez, "Robust image registration via cepstral analysis," in *2011 International Conference on Digital Image Computing: Techniques and Applications*, IEEE, 2011, pp. 45–50. DOI: 10.1109/DICTA.2011.16.

[50] J. Sarvaiya, S. Patnaik, and K. Kothari, "Image registration using log polar transform and phase correlation to recover higher scale," *Journal of Pattern Recognition Research*, pp. 90–105, 2012. DOI: 10.13176/11.355.

[51] J. Ren, T. Vlachos, Y. Zhang, J. Zheng, and J. Jiang, "Gradient-based subspace phase correlation for fast and effective image alignment," *Journal of Visual Communication and Image Representation*, pp. 1558–1565, 2014. DOI: 10.1016/j.jvcir.2014.07.001.

[52] H. Foroosh, J. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," *IEEE Transactions on Image Processing*, pp. 188–200, 2002. DOI: 10.1109/83.988953.

[53] Jinchang Ren, Jianmin Jiang, and T. Vlachos, "High-accuracy sub-pixel motion estimation from noisy images in fourier domain," *IEEE Transactions on Image Processing*, pp. 1379–1384, 2010. DOI: 10.1109/TIP.2009.2039056.

[54] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms.," *Optics letters*, pp. 156–158, 2008, ISSN: 0146-9592. DOI: 10.1364/OL.33.000156.

[55] S Baker and I Matthews, "Equivalence and efficiency of image alignment algorithms," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE, 2001, pp. I–1090–I–1097. DOI: 10.1109/CVPR.2001.990652.

[56] S. Baker and I. Matthews, "Lucas-kanade 20 years on: a unifying framework," *International Journal of Computer Vision*, pp. 221–255, 2004. DOI: 10.1023/B:VISI.0000011205.11775.fd.

[57] S. Benhimane and E. Malis, "Real-time image-based tracking of planes using efficient second-order minimization," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2004, pp. 943–948. DOI: 10.1109/IROS.2004.1389474.

[58] R. Brooks and T. Arbel, "Generalizing inverse compositional and esm image alignment," *International Journal of Computer Vision*, pp. 191–212, 2010. DOI: 10.1007/s11263-009-0263-8.

[59] N. Dowson and R. Bowden, "Mutual information for lucas-kanade tracking (milk): an inverse compositional formulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 180–185, 2008. DOI: 10.1109/TPAMI.2007.70757.

[60] A. Dame and E. Marchand, "Second-order optimization of mutual information for real-time image registration," *IEEE Transactions on Image Processing*, pp. 4190–4203, 2012. DOI: 10.1109/TIP.2012.2199124.

[61] M. R. Pickering, Y. Xiao, and X. Jia, "Registration of multi-sensor remote sensing imagery by gradient-based optimization of cross-cumulative residual entropy," in *SPIE Defense and Security Symposium*, International Society for Optics and Photonics, 2008, 69660U–69660U. DOI: 10.1117/12.777016.

[62] M. Hasan, M. R. Pickering, and X. Jia, "Robust automatic registration of multimodal satellite images using ccre with partial volume interpolation," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 4050–4061, 2012. DOI: 10.1109/TGRS.2012.2187456.

[63] G. G. Scandaroli, M. Meilland, and R. Richa, "Improving ncc-based direct visual tracking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, pp. 442–455. DOI: 10.1007/978-3-642-33783-3_32.

[64] J. P. Lewis, "Fast normalized cross-correlation," in *Vision Interface*, vol. 10, 1995, pp. 120–123.

[65] O. Mendoza-schrock, J. A. Patrick, and M. Garing, "Exploring image registration techniques for layered sensing," pp. 1–15, 2009. DOI: 10.1117/12.818526.

[66] O. Mendoza-Schrock, J. A. Patrick, and E. P. Blasch, "Video image registration evaluation for a layered sensing environment," in *National Aerospace and Electronics Conference, Proceedings of the IEEE*, 2009, pp. 223–230, ISBN: 9781424444946. DOI: 10.1109/NAECON.2009.5426624.

[67] A. Singh and M. Jagersand, "Modular tracking framework: a unified approach to registration based tracking," 2016. arXiv: 1602.09130.

[68] A. Singh, A. Roy, X. Zhang, and M. Jagersand, "Modular decomposition and analysis of registration based trackers," 2016. arXiv: 1603.01292.

[69] G. Wolberg and S. Zokai, "Robust image registration using log-polar transform," *Proceedings 2000 International Conference on Image Processing*, vol. 1, pp. 493–496, 2000. DOI: 10.1109/ICIP.2000.901003.

[70] P. N. Crabtree, C. Seanor, J. Murray-Krezan, and P. J. McNicholl, "Robust global image registration based on a hybrid algorithm combining fourier and spatial domain techniques," in *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, 2012.

[71] K. S. Jackovitz, "Integrated coarse to fine and shot break detection approach for fast and efficient registration of aerial image sequences," PhD thesis, University of Dayton, 2013. [Online]. Available: http://rave.ohiolink.edu/etdc/view?acc_num=dayton1366306702.

[72] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," *IEEE Transactions on Visualization and Computer Graphics*, pp. 355–368, 2010. DOI: 10.1109/TVCG.2009.99.

[73] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," *ACM Transactions on Graphics*, pp. 1–10, 2011. arXiv: arXiv:1204.6216v2.

[74] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Calibration-free rolling shutter removal," *2012 IEEE International Conference on Computational Photography ICCP*, pp. 1–8, 2012. DOI: 10.1109/ICCPhot.2012.6215213.

[75] E. Ringaby and P.-E. Forssén, "Efficient video rectification and stabilisation for cell-phones," *International Journal of Computer Vision*, pp. 335–352, 2012. DOI: 10.1007/s11263-011-0465-8.

[76] M. C. Yip, D. G. Lowe, S. E. Salcudean, R. N. Rohling, and C. Y. Nguan, "Tissue tracking and registration for image-guided surgery," *IEEE Transactions on Medical Imaging*, pp. 2169–2182, 2012. DOI: 10.1109/TMI.2012.2212718.

[77] M. Veldandi, S. Ukil, and K. G. Rao, "Video stabilization by estimation of similarity transformation from integral projections," in *2013 IEEE International Conference on Image Processing*, IEEE, 2013, pp. 785–789. DOI: 10.1109/ICIP.2013.6738162.

[78] I. E. Abdou, "Practical approach to the registration of multiple video images," *SPIE*, pp. 371–382, 1999. DOI: 10.1117/12.334685.

[79] Q. Tian and M. N. Huhns, "Algorithms for subpixel registration," in *Computer Vision, Graphics, and Image Processing*, 1986, pp. 220–233. DOI: 10.1016/0734-189X(86)90028-9.

[80] R. Bracewell, K.-Y. Chang, A. Jha, and Y.-H. Wang, "Affine theorem for two-dimensional fourier transform," *Electronics Letters*, p. 304, 1993. DOI: 10.1049/el:19930207.

# Appendix A

# 2D Fourier Transform

The definitions of the continuous and discrete Fourier transform are provided below. Following these definitions, four properties of the Fourier transform that are relevant to the hybrid registration method presented in Chapter 3 are derived.

## A.1 Definition of 2D Fourier Transform

The 2D Fourier transform is used to transform a 2D function or image from the spatial domain into a frequency domain representation consisting of a coefficient weighted sum of sine and cosine basis functions over regularly spaced frequencies where each frequency has both a magnitude and phase component. The underlying theory is based on the continuous Fourier transform of an infinite-length signal, but much of this theory also applies to the discrete Fourier transform of finite-length signals or images. The continuous forwards Fourier transform is defined as:

$$F(\omega_x, \omega_y) = \mathcal{F}(f(x,y)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i2\pi(\omega_x x+\omega_y y)}\,dx\,dy \tag{A.1}$$

where $\mathcal{F}$ denotes the Fourier transform, $x$,$y$ are coordinates in image space, $\omega_x$,$\omega_y$ are coordinates in the frequency domain and the relation to basis functions is given by:

$$e^{-i2\pi(\omega_x x+\omega_y y)} = \cos 2\pi(\omega_x x + \omega_y y) - i \sin 2\pi(\omega_x x + \omega_y y) \tag{A.2}$$

Similarly, the continuous inverse Fourier transform is defined as:

$$f(x,y) = \mathcal{F}^{-1}(F(\omega_x,\omega_y)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\omega_x,\omega_y)e^{i2\pi(\omega_x x+\omega_y y)}\,d\omega_x\,d\omega_y \tag{A.3}$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform and

$$e^{i2\pi(\omega_x x+\omega_y y)} = \cos 2\pi(\omega_x x+\omega_y y) + i\sin 2\pi(\omega_x x+\omega_y y) \tag{A.4}$$

For the case of 2D images, which are finite and discretely sampled, the discrete Fourier transform (DFT) must be used. The forwards DFT of a finite $M$x$N$ image is given by:

$$F(\omega_x,\omega_y) = \frac{1}{MN}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)e^{-i2\pi(\omega_x x/N+\omega_y y/M)} \tag{A.5}$$

Similarly, the inverse DFT is given by:

$$f(x,y) = \sum_{\omega_x=0}^{N-1}\sum_{\omega_y=0}^{M-1} F(\omega_x,\omega_y)e^{i2\pi(\omega_x x+\omega_y y)} \tag{A.6}$$

Different DFT implementations exist, the most common of which is the Fast Fourier Transform (FFT). Given square images, the computational complexity of the 2D FFT is $O(N^2\log N)$ compared to $O(N^3)$ for the naive DFT. As its name suggests, the FFT is a fast algorithm for computing the DFT and numerous highly optimized implementations are available.

## A.2 Affine Property of 2D Fourier Transform

The shift, similarity and rotation properties of the 2D Fourier transform can be generalized to a 2D affine property as shown and derived in [80]. This property determines the affect on the frequency domain result of the Fourier transform when an affine transformation is applied in the spatial domain. The derivation provided here is based on [80], but here matrix

notation is maintained throughout the derivation. First, consider the 2D Fourier transform in vectorized form:

$$\mathcal{F}(f(\begin{smallmatrix} x \\ y \end{smallmatrix})) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\begin{smallmatrix} x \\ y \end{smallmatrix}) e^{-i2\pi(\omega_x \ \omega_y)(\begin{smallmatrix} x \\ y \end{smallmatrix})} \, dx \, dy \tag{A.7}$$

where $x, y$ are spatial domain coordinates, $\omega_x, \omega_y$ are frequency domain coordinates, $f(\begin{smallmatrix} x \\ y \end{smallmatrix})$ is the 2D image function, and $\mathcal{F}(f(\begin{smallmatrix} x \\ y \end{smallmatrix}))$ is the Fourier transform of $f(\begin{smallmatrix} x \\ y \end{smallmatrix})$.

Let the spatial affine transformation be represented in matrix notation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{A.8}$$

where $x, y$ are the original spatial coordinates, $x', y'$ are the transformed spatial coordinates, $a, b, c, d$ are the affine parameters that account for rotation, scale, and shear, and $t_x, t_y$ account for translation. After rearranging we obtain:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} \begin{bmatrix} x' - t_x \\ y' - t_y \end{bmatrix} \tag{A.9}$$

Based on the relation between Jacobians in 2D:

$$dx' \, dy' = |\Delta| \, dx \, dy \tag{A.10}$$

where the determinant $\Delta$ is given by:

$$\Delta = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

By substituting Eq. (A.9) and Eq. (A.10) into Eq. (A.7):

$$\mathcal{F}(f(\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix})) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\begin{pmatrix} x' \\ y' \end{pmatrix}) e^{-i2\pi(\omega_x\ \omega_y)\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1}\begin{pmatrix} x'-t_x \\ y'-t_y \end{pmatrix}} dx'\,dy'/|\Delta| \tag{A.11}$$

The term $(\omega_x\ \omega_y)\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1}\begin{pmatrix} x'-t_x \\ y'-t_y \end{pmatrix}$ in the phase component can be simplified as follows:

$$\begin{aligned}(\omega_x\ \omega_y)\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1}\begin{pmatrix} x'-t_x \\ y'-t_y \end{pmatrix} &= (\omega_x\ \omega_y)\frac{1}{\Delta}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} x'-t_x \\ y'-t_y \end{pmatrix} \\ &= \frac{1}{|\Delta|}(\omega_x\ \omega_y)\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} x' \\ y' \end{pmatrix} - \frac{1}{|\Delta|}(\omega_x\ \omega_y)\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} t_x \\ t_y \end{pmatrix}\end{aligned}$$

Plugging this simplified phase component back into Eq. (A.11) and moving the constant part of the phase outside the integral gives:

$$\begin{aligned}&\mathcal{F}(f(\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix})) \\ &= e^{i2\pi/\Delta(\omega_x\ \omega_y)\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} t_x \\ t_y \end{pmatrix}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\begin{pmatrix} x' \\ y' \end{pmatrix}) e^{-i2\pi/\Delta(\omega_x\ \omega_y)\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} x' \\ y' \end{pmatrix}} dx'\,dy'/|\Delta|\end{aligned}$$

By definition of the Fourier transform:

$$\mathcal{F}(f(\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix})) = \frac{1}{\Delta}e^{i2\pi/\Delta(\omega_x\ \omega_y)\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} t_x \\ t_y \end{pmatrix}} F\left(\begin{pmatrix} d & -c \\ -b & a \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix}/|\Delta|\right)$$

where $F(\omega_x', \omega_y')$ is the Fourier transformed image. Finally, if we let $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, this can be further simplified to:

$$\mathcal{F}(f(M\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix})) = \frac{1}{\Delta}e^{i2\pi/\Delta(\omega_x\ \omega_y)M^{-1}\begin{pmatrix} t_x \\ t_y \end{pmatrix}} F\left(M^{-T}\begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix}/|\Delta|\right) \tag{A.12}$$

which shows the relationship between transformed and original frequency domain coordinates when an affine transformation as shown in Eq. (A.8) has been applied to the spatial coordinates. Note that there is a global change in magnitude by a factor of $1/\Delta$ and the phase change is a function of all 6 affine parameters. The corresponding transformation to

frequency domain coordinates is:

$$\begin{pmatrix} \omega'_x \\ \omega'_y \end{pmatrix} = M^{-T} \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix} / |\Delta| \tag{A.13}$$

where $\omega_x, \omega_y$ are the original frequency domain coordinates and $\omega'_x, \omega'_y$ are the transformed frequency domain coordinates resulting from the affine transformation applied in the spatial domain. By looking at subsets of affine transformations, such as rotation, scale, and translation, it is straight forward to derive the corresponding Fourier property from Eq. (A.12).

## A.3 Rotation Property of 2D Fourier Transform

The derivation of the rotation property for the Fourier transform of a 2D function or image follows naturally from Eq. (A.12) above. For an image function $f(x, y)$ spatially rotated by angle $\theta$, the transformation matrix $M$ is:

$$M = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{A.14}$$

Substituting Eq. (A.14) into Eq. (A.12) and noting that $(t_x\ t_y)^T = (0\ 0)^T$ results in:

$$\mathcal{F}(f(\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix})) = \frac{1}{\Delta} F\left(\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix} / |\Delta|\right) \tag{A.15}$$

Noting that the determinant of $M$

$$\Delta = \begin{vmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{vmatrix} = \cos^2\theta + \sin^2\theta = 1 \tag{A.16}$$

allows Eq. (A.15) to be simplified to:

$$\mathcal{F}(f(\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix})) = F\left(\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix}\right) \tag{A.17}$$

Alternatively, Eq. (A.17) can be represented in non-vectorized form:

$$\mathcal{F}(f(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)) = F(\omega_x\cos\theta + \omega_y\sin\theta, -\omega_x\sin\theta + \omega_y\cos\theta) \tag{A.18}$$

where it is also apparent that rotation in the spatial domain corresponds to rotation by the same angle of the spectrum magnitude in the frequency domain.

## A.4 Similarity Property of 2D Fourier Transform

The derivation of the similarity property for the Fourier transform of a 2D function or image follows naturally from Eq. (A.12) above. For an image function $f(x, y)$ spatially scaled by positive scale $a$, the transformation matrix $M$ is:

$$M = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix} \tag{A.19}$$

Substituting Eq. (A.19) into Eq. (A.12) and noting that $(t_x\ t_y)^T = (0\ 0)^T$ results in:

$$\mathcal{F}(f(\left(\begin{smallmatrix} a & 0 \\ 0 & a \end{smallmatrix}\right)\left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right))) = \frac{1}{\Delta}F\left(\left(\begin{smallmatrix} a & 0 \\ 0 & a \end{smallmatrix}\right)\left(\begin{smallmatrix} \omega_x \\ \omega_y \end{smallmatrix}\right)/|\Delta|\right) \tag{A.20}$$

Noting that the determinant of $M$

$$\Delta = a^2$$

allows Eq. (A.20) to be simplified to:

$$\mathcal{F}(f(\left(\begin{smallmatrix} a & 0 \\ 0 & a \end{smallmatrix}\right)\left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right))) = \frac{1}{a^2}F\left(\left(\begin{smallmatrix} 1/a & 0 \\ 0 & 1/a \end{smallmatrix}\right)\left(\begin{smallmatrix} \omega_x \\ \omega_y \end{smallmatrix}\right)\right) \tag{A.21}$$

Alternatively, Eq. (A.21) can be represented in non-vectorized form:

$$\mathcal{F}(f(ax, ay)) = \frac{1}{a^2}F(\frac{\omega_x}{a}, \frac{\omega_y}{a}) \tag{A.22}$$

where it is clear that scale in the spatial domain corresponds to inverse scale of the spectrum magnitude in the frequency domain.

## A.5 Shift Property of 2D Fourier Transform

The derivation of the similarity property for the Fourier transform of a 2D function or image follows directly from Eq. (A.12) above. For an image function $f(x, y)$ that is spatially translated, the transformation matrix $M$ is:

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.23}$$

Substituting Eq. (A.23) into Eq. (A.12) and noting that $(t_x\ t_y)^T$ is nonzero results in:

$$\mathcal{F}(f(\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix})) = e^{i2\pi(\omega_x\ \omega_y)\begin{pmatrix} t_x \\ t_y \end{pmatrix}} F\begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix} \tag{A.24}$$

Alternatively, Eq. (A.24) can be represented in non-vectorized form:

$$\mathcal{F}(f(a + t_x, y + t_y)) = e^{i2\pi(\omega_x t_x + \omega_y t_y)} F(\omega_x, \omega_y) \tag{A.25}$$

where it can be observed that a translational shift in the spatial domain corresponds to a phase shift in the frequency domain with no impact on the spectrum magnitude.

# Appendix B

# Proposed Hybrid Coarse-Fine Method Pseudocode

Pseudocode for the coarse and fine methods that are used in the proposed hybrid method is provided below. The procedure naming convention is consistent with the steps used and described in detail under the methods in Chapter 3, specifically in Fig. 3.2, Fig. 3.3, and Fig. 3.9. Where appropriate, references to respective equations are provided in comments throughout the pseudocode.

## B.1 Coarse Registration Method Pseudocode

Pseudocode is provided below for the complete coarse registration method, including NGC, OC, and the scale space search.

**Algorithm 1:** RegisterCoarse

**Input**: $I_{src}$, $I_{dst}$, $p_{rs}$, $p_t$, $N_\theta$, $N_s$, $PSR_{threshold}$
**Output**: $RST(\theta, s, t_x, t_y)$, returns $true$ if successful and $false$ otherwise

```
// build Gaussian Pyramids
```
$p_{max} \leftarrow \max(p_{rs}, p_t)$
$P_{src} \leftarrow$ `GaussianPyramid`$(I_{src}, p_{max})$
$P_{dst} \leftarrow$ `GaussianPyramid`$(I_{dst}, p_{max})$

```
// search over initial scale estimates
```
$peak\_max \leftarrow 0$
**foreach** $s_{est} \in \{1, 2, \frac{1}{2}, 4, \frac{1}{4}, 6, \frac{1}{6}\}$ **do**

```
    // estimate rotation,scale using NGC
```
    $RS(\theta, s) \leftarrow$ `RegisterNGC`$(P_{src}, P_{dst}, p_{rs}, s_{est}, N_\theta, N_s)$

```
    // estimate translation using OC
```
    $T(t_x, t_y), peak, sucess \leftarrow$ `RegisterOC`$(P_{src}, P_{dst}, p_t, RS(\theta, s), PSR_{threshold})$

    **if** $success$ **and** $peak > peak\_max$ **then**
        $peak\_max \leftarrow peak$
        $RST(\theta, s, t_x, t_y) \leftarrow T(t_x, t_y) * RS(\theta, s)$
    **end**
**end**
**return** $peak\_max > 0$

**Algorithm 2:** RegisterNGC

**Input**: $P_{src}, P_{dst}, p_{rs}, s_{est}, N_\theta, N_s$
**Output**: $RS(\theta, s)$

```
// select appropriate pyramid based on scale
```
$P_1 \leftarrow P_{src}$
$P_2 \leftarrow P_{dst}$
**if** $s_{est} < 1$ **then** $P_1 \leftrightarrow P_2$

```
// calculate pyramid levels, select and crop images
// according to Eq. (3.12)
```
$n \leftarrow p_{rs} - |\log_2(s_{est})|$
$p_1 \leftarrow \max(n, 0)$
$p_2 \leftarrow p_{rs}$
$crop\_size \leftarrow 2^{\min(n,0)-p_{rs}}$
$I_1 \leftarrow \texttt{Crop}(P_1[p_1], crop\_size)$
$I_2 \leftarrow P_2[p_2]$

```
// pad smaller image if necessary
```
$I_1 \leftarrow \texttt{Pad}(I_1, Size(I_2))$

```
// compute complex gradients using Eq. (3.1)
```
$G_1 \leftarrow \texttt{ComplexGradient}(I_1)$
$G_2 \leftarrow \texttt{ComplexGradient}(I_2)$

```
// compute spectrum magnitudes of complex gradients
// by applying FMT in Eq. (2.10)
```
$M_1 \leftarrow \texttt{FMT}(G_1)$
$M_2 \leftarrow \texttt{FMT}(G_2)$

```
// apply log polar transform using Eq. (3.2)
```
$M_1 \leftarrow \texttt{LPT}(M_1)$
$M_2 \leftarrow \texttt{LPT}(M_2)$

```
// compute NGC using Eq. (3.3)
```
$C_{ngc} \leftarrow \texttt{NGC}(ComplexGradient(M_1), ComplexGradient(M_2))$

```
// find NGC max peak
```
$peak_x, peak_y \leftarrow \texttt{FindMax}(C_{ngc})$

```
// perform subpixel peak fit using Eq. (3.5)
```
$peak_x, peak_y \leftarrow \texttt{SubpixelGaussianPeakFit}(C_{ngc}, peak_x, peak_y)$

```
// compute rotation and scale using Eq. (3.6)
```
$\theta, s \leftarrow \texttt{ComputeRotationScale}(peak_x, peak_y, N_\theta, N_s)$

```
// update computed scale based on scale estimate
```
$s \leftarrow s * s_{est}$
**if** $s_{est} < 1$ **then** $s \leftarrow s^{-1}$

**Algorithm 3:** RegisterOC

**Input**: $P_{src}$, $P_{dst}$, $p_t$, $RS(\theta, s)$, $PSR_{threshold}$
**Output**: $T(t_x, t_y)$, $max\_peak$, returns $true$ if successful and $false$ otherwise

```
// select appropriate pyramid based on scale
```
$P_1 \leftarrow P_{src}$; $P_2 \leftarrow P_{dst}$
**if** $s > 1$ **then** $s_{effective} \leftarrow s^{-1}$; $P_1 \leftrightarrow P_2$
**else** $s_{effective} \leftarrow s$

```
// select images
```
$I_1 \leftarrow P_1[p_t]$; $I_2 \leftarrow P_2[p_t]$

```
// test θ and θ + 180 to resolve rotation ambiguity
```
$peak\_max \leftarrow 0$
**for** $\theta_{effective} \leftarrow \{\theta, \theta + 180\}$ **do**

```
    // rotation/scale compensation: Eq. (2.3), Eq. (2.4)
```
    $I_1 \leftarrow Warp(I_1, RS(\theta_{effective}, s_{effective}))$

```
    // pad I2 to be same size as I1
```
    $I_2 \leftarrow Pad(I_2, Size(I_1))$

```
    // compute complex gradient orientation: Eq. (3.7)
```
    $G_1 \leftarrow ComplexGradientOrientation(I_1)$
    $G_2 \leftarrow ComplexGradientOrientation(I_2)$

```
    // compute OC using Eq. (3.8)
```
    $C_{oc} \leftarrow OC(G_1, G_2)$

```
    // smooth OC surface
```
    $C_{oc} \leftarrow Conv(C_{oc}, GaussianKernel_{5x5}(\sigma = 1.0))$

```
    // find OC max peak
```
    $peak, peak_x, peak_y \leftarrow FindMax(C_{ngc})$

```
    // PSR threshold test using Eq. (3.9)
```
    **if** $PSR(C_{oc}, peak_x, peak_y) > PSR_{threshold}$ **and** $peak > peak\_max$ **then**
        $peak\_max \leftarrow peak$
        $peak\_max_x \leftarrow peak_x$; $peak\_max_y \leftarrow peak_y$
    **end**
**end**
**if** $peak\_max$ **is** 0 **then return** *false*

```
// perform subpixel peak fit using Eq. (3.10)
```
$peak\_max_x, peak\_max_y \leftarrow SubpixelGaussianPeakFit(C_{oc}, peak\_max_x, peak\_max_y)$

```
// compute translation using Eq. (3.11)
```
$t_x, t_y \leftarrow ComputeTranslation(peak\_max_x, peak\_max_y)$

```
// update translation based on scale estimate
```
**if** $s > 1$ **then** $t_x \leftarrow -t_x$; $t_y \leftarrow -t_y$

**return** *true*

## B.2 Fine Registration Method Pseudocode

Pseudocode for the fine registration method using grid-based NCC is provided below.

**Algorithm 4:** RegisterFine

**Input**: $I_{src}, I_{dst}, template_w, window_w, grid_x, grid_y, min\_inliers$
**Output**: $H$, returns $true$ if successful and $false$ otherwise

```
// order images so I_1 will be scaled up to match I_2
```
$I_1 \leftarrow I_{src}$
$I_2 \leftarrow I_{dst}$
**if** $s_{est} < 1$ **then** $P_1 \leftrightarrow P_2$
$w, h \leftarrow$ `Size`$(I_1)$
$template_{hw} \leftarrow \lfloor template_w/2 \rfloor$
$window_{hw} \leftarrow \lfloor window_w/2 \rfloor$

```
// compute grid locations
```
$\mathsf{x}, \mathsf{y} \leftarrow$ `Grid`$(w, h, grid_x, grid_y)$

```
// for each grid location
```
**for** $i \leftarrow 0; i <$ `Length`$(\mathsf{x}); i \leftarrow i + 1$ **do**

```
    // window center location
```
    $cw_x \leftarrow \lfloor \mathsf{x}[i] + (grid_w - 1)/2 \rfloor$
    $cw_y \leftarrow \lfloor \mathsf{y}[i] + (grid_h - 1)/2 \rfloor$

```
    // ignore template outside image bounds
```
    **if** $cw_x + template_{hw} \geq w$ **or** $cw_y + template_{hw} \geq h$ **or** $cw_x - template_{hw} < 0$ **or** $cw_y - template_{hw} < 0$ **then continue**

```
    // select template region
```
    $template \leftarrow I_1($`Range`$(cw_x - template_w, cw_x + template_w + 1),$ `Range`$(cw_y - template_w, cw_y + template_w + 1))$

```
    // select search window region
```
    $startw_x, endw_x, startw_y, endw_y \leftarrow$ `WindowExtents`$(cw_x, cw_y, w, h, window_{hw})$
    $window \leftarrow I_2($`Range`$(startw_x, endw_x + 1),$ `Range`$(startw_y, endw_y + 1))$

```
    // count total and non-zero pixels in template/window
```
    $nb, nw, zb, zw \leftarrow$ `PixelCount`$(window, template, window_w, template_w)$

```
    // non-uniform templates with more than half non-zero pixels
```
    **if** `StandardDeviation`$(template) > 0$ **and** $zb/nb < 0.5$ **and** $zw/nw < 0.5$ **then**

```
        // compute NCC using Eq. (3.13)
```
        $C_{ncc} \leftarrow$ `NormalizedCrossCorrelation`$(window, template)$

```
        // find NCC max peak
```
        $peak_x, peak_y \leftarrow$ `FindMax`$(C_{ncc})$

```
        // perform subpixel peak fit using Eq. (3.15)
```
        $peak_x, peak_y \leftarrow$ `SubpixelQuadraticPeakFit`$(C_{ncc}, peak_x, peak_y)$

```
        // compute x,y locations in input image coordinates
```
        $(x_1[i], y_1[i]) \leftarrow (cw_x, cw_y)$
        $(x_2[i], y_2[i]) \leftarrow (peak_x + block_{hw} + startw_x, peak_y + block_{hw} + startw_y)$
    **end**
**end**
$H, success \leftarrow$ `ComputeHomography`$(x_1, y_1, x_2, y_2, min\_inliers)$
**if** $s_{est} < 1$ **then** $H \leftarrow H^{-1}$

**return** $success$

**Procedure** Grid($w, h, grid_x, grid_y$)

**Output**: x, y
$gx \leftarrow w/grid_x$
$gy \leftarrow h/grid_y$
$k \leftarrow 0$
**for** $i \leftarrow 0; i < h-1; i \leftarrow i+gy$ **do**
  **for** $j \leftarrow 0; j < w-1; j \leftarrow j+gx$ **do**
    x[$k$] $\leftarrow j$
    y[$k$] $\leftarrow i$
    $k \leftarrow k+1$
  **end**
**end**

**Procedure** WindowExtents($cw_x$, $cw_y$, $w$, $h$, $window_{hw}$)

**Output**: $startw_x$,$endw_x$,$startw_y$,$endw_y$
$startw_x \leftarrow \max(0, cw_x - window_{hw})$
$endw_x \leftarrow \min(w-1, cw_x + window_{hw})$
$startw_y \leftarrow \max(0, cw_y - window_{hw})$
$endw_y \leftarrow \min(h-1, cw_y + window_{hw})$

**Procedure** PixelCount($window$, $template$, $window_w$, $template_w$)

**Output**: $nb$, $nw$, $zb$, $zw$
// count number of pixels in template/window
$nb \leftarrow template_w * template_w$
$nw \leftarrow window_w * window_w$
// count number of zero valued pixels in template/window
$zb \leftarrow nb -$ CountNonZero($template$)
$zw \leftarrow nw -$ CountNonZero($window$)

**Procedure** ComputeHomomgraphy($x_1, y_1, x_2, y_2, min\_inliers$)

**Output**: $H$, returns $true$ if successful and $false$ otherwise
// remove outliers using RANSAC
$H, num\_inliers \leftarrow$ RANSACHomography($x_1, y_1, x_2, y_2$)
**if** $num\_inliers \geq min\_inliers$ **then return** $true$
**else return** $false$